INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
TRADE AND WAR IN CELLULAR AUTOMATA WORLDS:
A COMPUTER SIMULATION OF INTERSTATE INTERACTIONS

DISSERTATION

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the Graduate
School of The Ohio State University

By
Byoung Won Min, M.A.

The Ohio State University
2002

Dissertation Committee:
Professor Brian M. Pollins, Advisor
Professor Donald A. Sylvan
Professor Richard J. Timpone

Approved by
[Signature]
Brian M. [Name]
Department of Political Science

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
ABSTRACT

This study investigated the impact of the micro-level factors of state agents on the pattern of the system-level properties in international relations, and the impact of the dyadic and systemic factors on the probability of war, with the computer simulation method. Considering that existing simulation models in international relations have focused on the conflict dimension, this study tried to integrate the cooperative dimension of world politics in order to see what would happen in the integrated model. Computer simulations, using the cellular automata framework and the complex systems theory, were designed to generate artificial datasets for testing the nine propositions related to major international relations theories. In particular, this study combined Cusack and Stoll’s conflict model and Epstein and Axtell’s trade model into one in order to experiment with both the realist and the liberal elements of world politics – war and trade.

Three major tasks were undertaken for the main topic of this study. First, a replicated model of security-only world was compared to a newly integrated model of the security-and-trade world. Three dependent variable sets – system endurance, balance-of-power, and state survival rate – were compared between these two worlds. The introduction of trade into the model made system endurance more probable than the security-only world, especially in terms of the iteration numbers. The probability of
deterrent balance-of-power increased in the security-and-trade world, while there was little difference in the probability of defensive balance-of-power. State survival rate increased with the trade option, but system-level factors had more impact than state-level factors in the security-and-trade world.

Second, the integrated model of war and trade was also used to test the relationship between dyadic factors – economic interdependence and heterogeneity – to the probability of war. In this cellular automata world, the increase of economic interdependence reduced the probability of war. The type of dyads, measured by the heterogeneity in state power, had a great impact on the probability of war, too. Wars happened the most frequently in the minor-minor dyad, while major states were more prudent in starting war.

The third task concerned the impact of systemic factors – polarity and power concentration – on the probability of war. Multipolar worlds were the most war-prone, while unipolar worlds were the least. A similar result appeared with the number of major powers, instead polarity, as the explanatory variable. Also the power concentration among major powers was found to show an inverted U-shaped curve with the probability of war lowest when the number of great powers was two or three.
To my wife, Hyesook

iv
ACKNOWLEDGMENTS

Many people have been involved in my project for this dissertation, but I have to mention the names of my dissertation committee members first: Professors Brian Pollins, Donald Sylvan, and Richard Timpone. Professor Pollins has been the axis during my long career at the Ohio State University. He suggested many guidelines to my research and pulled me back in the field of political science whenever I was wandering around peripheries. He always made me admire his acuteness whenever I discussed models, theories and research themes regarding my dissertation. Professor Sylvan has been a mentor to me when I needed him: I had to rely on his advice and help when I felt frustrated by many unfavorable situations. He was rigorous in his class, but he was generous in social life. I did not have any courses from Professor Timpone, but his help in the final stage of my dissertation was indispensable: he suggested a lot of valuable points to my simulation models and statistical analyses. Without his help, this dissertation might not be possible.

In relation to my research project, I would like to express my gratitude to two research institutions. The first is the Center for Integrated Design at the Biomedical Center of the Ohio State University. In particular, my experience with Professor Manjula Waldron was special. She introduced the complex adaptive systems (CAS) theory to me.
which changed my research agenda from the conventional international relations theories to a newly emerging paradigm. Dr. Ron Eglash, Brian Clouse, Raja Laifa, Cindy Spreng, Jun Liu, Zhifeng Wang, and Jayshree Radhakrishnan were friendly colleagues at the Center who shared resources and ideas on the CAS for two years. The other institution was the Mershon Center which funded my research on the power law distribution of international conflicts upon the CAS paradigm. I would like to appreciate Dr. Ned Lebow and Professor Pollins' support at the Mershon Center. Also I thank Professor Richard Hamilton, who always inspired me with his extraordinary insights on many political and social matters.

Many other scholars have contributed to the development of my final product. Professor Richard Stoll at Rice University provided me with his valuable program source code of his original model for my research, in addition to his interest and critical comments. Professors Gavan Duffy and Lars-Erik Cederman were vital resources for the computer simulation design of my research: they offered me some precious guidelines on parallel computing and the agent-based modeling framework. Professors Dina Zinnes, Lisa Martin, Jack Snyder, Volker Krause, Harvard Hegre, Hayward Alker, Stephen Majeski, Robert Axtell, Doh C. Shin, and Taehyun Kim also suggested many valuable comments on my projects at seminars and conferences. Mr. Gary Stroup was an excellent outside reader of my dissertation who corrected many errors in my English. Finally I would like to thank my wife, Hyesook, and my lovely twin sons, Hyoungki and Wanki, for their long journey with me.
VITA

August 22, 1963 ............................... Born – Youngwol, Republic of Korea

1988 ............................................... M.A. Political Science,
Seoul National University,
Republic of Korea

1988 – 1993 ................................... Research Associate
The Sejong Institute, Republic of Korea

1997 – 1999 ................................. Honda/Ford Research Fellow
The Center for Integrated Design
The Ohio State University

1999 – 2001 ................................. Graduate Research Associate
The Mershon Center for International
Security and Public Policy
The Ohio State University

vii
PUBLICATIONS


FIELDS OF STUDY

Major Field: Political Science

1. Major: International Relations

   A. International Political Economy
   B. Foreign Policy

2. Minor: Comparative Politics
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Abstract</th>
<th>ii</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dedication</td>
<td>iv</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>v</td>
</tr>
<tr>
<td>Vita</td>
<td>vii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xiv</td>
</tr>
</tbody>
</table>

## Chapters:

1. Introduction ................................................................. 1
   1.1 Motivation ................................................................. 1
   1.2 Puzzles ................................................................. 4
   1.3 Description of Chapters ........................................... 12

2. Theory and Propositions ............................................... 15
   2.1 Theoretical Contexts and Literature Review ............... 15
      2.1.1 Trade and War as State Choices .......................... 16
      2.1.2 Interdependence and War ................................. 20
      2.1.3 Neorealism, Constructivism and International Structure ... 33
      2.1.4 Cellular Automata Simulations in International Relations ... 38
   2.2 Theoretical Propositions .......................................... 49
      2.2.1 War and Trade ............................................... 49
      2.2.2 Interdependence, Heterogeneity and War ............ 52
      2.2.3 Polarity, Power Concentration and War ............... 59
### 3. Research Methods

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Cellular Automata Simulations</td>
<td>67</td>
</tr>
<tr>
<td>3.2 Object-Oriented Programming and Agent-Based Modeling</td>
<td>74</td>
</tr>
<tr>
<td>3.3 Algorithms: War and Trade</td>
<td>78</td>
</tr>
<tr>
<td>3.3.1 Basic Module</td>
<td>78</td>
</tr>
<tr>
<td>3.3.2 War Module</td>
<td>89</td>
</tr>
<tr>
<td>3.3.3 Trade Module</td>
<td>92</td>
</tr>
<tr>
<td>3.4 Variables and Equations</td>
<td>95</td>
</tr>
</tbody>
</table>

### 4. Trade and War

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Research Themes</td>
<td>99</td>
</tr>
<tr>
<td>4.2 Research Design</td>
<td>102</td>
</tr>
<tr>
<td>4.2.1 Variables</td>
<td>102</td>
</tr>
<tr>
<td>4.2.2 Equations</td>
<td>108</td>
</tr>
<tr>
<td>4.3 Analysis of Results</td>
<td>118</td>
</tr>
<tr>
<td>4.3.1 System Endurance</td>
<td>118</td>
</tr>
<tr>
<td>4.3.2 Balance-of-Power</td>
<td>126</td>
</tr>
<tr>
<td>4.3.3 State Survival</td>
<td>134</td>
</tr>
<tr>
<td>4.4 Summary</td>
<td>140</td>
</tr>
</tbody>
</table>

### 5. Interdependence, Heterogeneity and War

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Research Themes</td>
<td>144</td>
</tr>
<tr>
<td>5.2 Research Design</td>
<td>146</td>
</tr>
<tr>
<td>5.2.1 Variables</td>
<td>146</td>
</tr>
<tr>
<td>5.2.2 Equations</td>
<td>152</td>
</tr>
<tr>
<td>5.3 Analysis of Results</td>
<td>154</td>
</tr>
<tr>
<td>5.3.1 Interdependence and War</td>
<td>154</td>
</tr>
<tr>
<td>5.3.2 Heterogeneity and War</td>
<td>159</td>
</tr>
<tr>
<td>5.4 Summary</td>
<td>170</td>
</tr>
</tbody>
</table>

### 6. Polarity, Power Concentration and War

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Research Themes</td>
<td>173</td>
</tr>
<tr>
<td>6.2 Research Design</td>
<td>175</td>
</tr>
<tr>
<td>6.2.1 Variables</td>
<td>175</td>
</tr>
<tr>
<td>6.2.2 Equations</td>
<td>183</td>
</tr>
<tr>
<td>6.3 Analysis of Results</td>
<td>185</td>
</tr>
<tr>
<td>6.3.1 Polarity and War</td>
<td>185</td>
</tr>
<tr>
<td>6.3.2 Power Concentration and War</td>
<td>195</td>
</tr>
<tr>
<td>6.4 Summary</td>
<td>200</td>
</tr>
</tbody>
</table>
7. Conclusions .............................................................................................................. 203

7.1 Propositions Revisited ..................................................................................... 204
7.2 Theoretical Implications ................................................................................. 206
   7.2.1 State and Choice ......................................................................................... 206
   7.2.2 Interdependence, Heterogeneity and War ................................................. 208
   7.2.3 Neorealism, Constructivism and Structural Factors .............................. 210
   7.2.4 Cellular Automata Simulations in International Relations ................... 213
7.3 Limitations of the Study ................................................................................... 217
7.4 Research Agenda for Future ........................................................................... 219

References .................................................................................................................. 223

Appendix 1: Relative Power Ratio of Major States, 1816-1965 ............................ 231

Appendix 2: Program Source Code in C++ ............................................................. 235
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>24</td>
</tr>
<tr>
<td>2.2</td>
<td>48</td>
</tr>
<tr>
<td>4.1</td>
<td>103</td>
</tr>
<tr>
<td>4.2</td>
<td>108</td>
</tr>
<tr>
<td>4.3</td>
<td>113</td>
</tr>
<tr>
<td>4.4</td>
<td>123</td>
</tr>
<tr>
<td>4.5</td>
<td>125</td>
</tr>
<tr>
<td>4.6</td>
<td>127</td>
</tr>
<tr>
<td>4.7</td>
<td>127</td>
</tr>
<tr>
<td>4.8</td>
<td>129</td>
</tr>
<tr>
<td>4.9</td>
<td>131</td>
</tr>
<tr>
<td>4.10</td>
<td>133</td>
</tr>
<tr>
<td>4.11</td>
<td>135</td>
</tr>
<tr>
<td>4.12</td>
<td>136</td>
</tr>
<tr>
<td>4.13</td>
<td>138</td>
</tr>
<tr>
<td>4.14</td>
<td>139</td>
</tr>
<tr>
<td>5.1</td>
<td>150</td>
</tr>
</tbody>
</table>
5.2 Interdependence Levels and War/Trade Ratio ........................................... 155
5.3 Impact of Economic Interdependence .......................................................... 158
5.4 Iteration Group and War/Trade Ratios ......................................................... 158
5.5 Heterogeneity and War Ratio ........................................................................ 160
5.6 Impact of Economic Interdependence in Major-Major Dyads .................... 162
5.7 Impact of Economic Interdependence in Major-Minor Dyads ...................... 162
5.8 Impact of Economic Interdependence in Minor-Major Dyads ..................... 163
5.9 Impact of Economic Interdependence in Minor-Minor Dyads ..................... 163
5.10 Wars in Major Power Dyads and Minor Power Dyads ............................... 164
5.11 Wars in Heterogeneous Dyads and Homogeneous Dyads ........................... 165
5.12 Impact of Power Ratio .................................................................................. 166
5.13 Impact of Power Ratio in Major-Major Dyads ............................................ 168
5.14 Impact of Power Ratio in Major-Minor Dyads ............................................ 168
5.15 Impact of Power Ratio in Minor-Major Dyads ............................................ 169
5.16 Impact of Power Ratio in Minor-Minor Dyads ............................................ 169
6.1 Polarity and the Probability of War .............................................................. 186
6.2 Impact of Polarity on War Probability ........................................................... 187
6.3 Number of Major States and the Probabilities of Trade and War ................ 189
6.4 Number of Major States and the Probability of War ................................... 191
6.5 Impact of Power Concentration on War Probability .................................... 199
7.1 Propositions and the Summary of Findings ............................................... 205
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Structure of Puzzles</td>
<td>11</td>
</tr>
<tr>
<td>2.1</td>
<td>Organization of Propositions</td>
<td>66</td>
</tr>
<tr>
<td>3.1</td>
<td>Cellular Automata Structure of World Politics</td>
<td>73</td>
</tr>
<tr>
<td>3.2</td>
<td>Cusack and Stoll's Basic Algorithm</td>
<td>83</td>
</tr>
<tr>
<td>3.3</td>
<td>Initiator Selection Process</td>
<td>84</td>
</tr>
<tr>
<td>3.4</td>
<td>Conflict Estimation Process</td>
<td>85</td>
</tr>
<tr>
<td>3.5</td>
<td>Relative Power and the Likelihood of Victory</td>
<td>90</td>
</tr>
<tr>
<td>3.6</td>
<td>Epstein and Axtell's Trade Algorithm</td>
<td>97</td>
</tr>
<tr>
<td>3.7</td>
<td>Combined Algorithm</td>
<td>98</td>
</tr>
<tr>
<td>4.1</td>
<td>System Endurance (Security-Only World, LV Fixed to 1.0)</td>
<td>119</td>
</tr>
<tr>
<td>4.2</td>
<td>System Endurance (Security-Only World, LV Varying)</td>
<td>120</td>
</tr>
<tr>
<td>4.3</td>
<td>System Endurance (Security-and-Trade World with 1,000 Iterations)</td>
<td>120</td>
</tr>
<tr>
<td>4.4</td>
<td>System Endurance (Security-and-Trade World with 2,000 Iterations)</td>
<td>121</td>
</tr>
<tr>
<td>4.5</td>
<td>System Endurance (Security-and-Trade World with 3,000 Iterations)</td>
<td>121</td>
</tr>
<tr>
<td>5.1</td>
<td>Distribution of Expected Utility</td>
<td>155</td>
</tr>
<tr>
<td>5.2</td>
<td>Heterogeneity in State Size</td>
<td>160</td>
</tr>
<tr>
<td>6.1</td>
<td>System Polarity Classification Algorithm</td>
<td>177</td>
</tr>
</tbody>
</table>
6.2 Power Configuration of Major States, 1816-1965 ........................................ 178
6.3 System CON, 1816-1965 ........................................................................ 182
6.4 Distribution of Polarity ........................................................................... 186
6.5 Distribution of the Number of Major States ........................................... 189
6.6 Number of Major States and the Probability of War ............................... 190
6.7 Interdependence and the Probability of War by Polarity ......................... 193
6.8 Power Ratio and the Probability of War by Polarity ................................. 194
6.9 Distribution of the Number of Major States (N > 1) ............................... 196
6.10 Power Concentration among Major States ............................................. 196
6.11 Power Concentration and the Probability of War ................................... 198
CHAPTER 1

INTRODUCTION

1.1 Motivation

This study is about modeling the conflict and cooperation in world politics with computer simulations. Specifically, it tries to build a computational model of war and trade between states with the essential elements of world politics. One of the most important motivations of this research is my interest in the combination of conflict and cooperative factors in explaining interstate relations. Two World Wars and the Cold War have epitomized the 20th century, during which the realist theory of power politics has dominated our perspective on world politics. Indeed, many practitioners and theorists tend to explain the process and the structure of world politics with the realist concepts such as anarchy, self-help, power, and conflict. One thing to note is that the realist story and scenario of international relations have been gloomy and pessimistic. However, even in the war-ridden 20th century, we have observed many counter-examples – cooperation among states – during the past decades. This study tries to focus on this dimension of cooperation in world politics.
Although the fundamental motivation of this study starts from an effort to understand the real world politics better, its main research orientation is toward modeling the process of international conflict and cooperation upon the basis of existing theories. The first task to be implemented in this study is to build a simple model of world politics with the essential elements extracted from existing international relations theories. In particular, I would like to model world politics with the computational method including algorithms and computer programs. The computational method is appropriate whenever the system processes are not tractable by the formal method or real-world data are limited or scarce for statistical inferences. Computer simulations in international relations have been used for decades but most have focused on the factor of power politics and conflict. In order to fill this gap, I would like to add the liberal factor – cooperation or trade – in modeling world politics and see what would happen with this addition. The past history of world politics needs not to be explained only by the pessimistic theory of realism: we have to explain many cooperative orders emerging out of the interactions among states. I want to experiment with the new model of conflict and cooperation vis-à-vis the conflict-centered model with computer simulations. Will they generate the same pattern of world politics at the macro-level? I would like to see whether these two models (with and without the liberal factor) generate any difference so that I can verify whether adding the liberal factors in building a model of world politics deserves any theoretical attention.

Two more extra tasks are intended in this study. Testing the relationship between war and economic interdependence is one of the tasks. Based upon the computational model built as the base framework of this study, I would like to test some theoretical propositions on this topic in my artificial world. Will this artificial world, composed only
of the essential elements suggested by major international relations theories, generate the same pattern between economic interdependence and conflict as that of the real world? With many empirical findings on this theme in mind, I would like to simulate it in my virtual world. This experiment with computer simulations may help to clarify the theoretical points in existing debate around this topic. In addition to this replication and test, I would like to see how the relationship (between economic interdependence and war) varies across different dyad types. If we try to understand the interactions among states with dyadic data, the property of dyads should be explored before our theoretical investigation of the main topic. For example, the dyad between superpowers and the dyad between small countries might not be the same in the pattern of conflict. I would like to see if there is any variation in them across different dyads.

Another theme to be tested with the base model in this study is the relationship between the structural factors – polarity and power concentration – and the probability of war in the system of states. The relationship between structure and agent has been a cross-paradigmatic topic between neorealism, neoliberal institutionalism, and constructivism. However, the direction of causality between these two factors is different in each theory. In the classical realist and the liberal traditions, the agent-level factors such as the nature of nation-state or domestic politics have been the primary sources of international relations. On the other hand, the structural variables are emphasized in the neorealist camp. The constructivist tradition takes the third approach by arguing that structure and agent are co-determined by the other. Upon these variances in theoretical traditions regarding the structure-agent relationship, I would like to see how the structural factors influence the state behavior, especially in the pattern of conflict.
In terms of methodology, this thesis calls attention to the limits of existing theories in terms of theoretical and empirical evidence, especially in the theory of economic interdependence and in the debate on the role of structural factors between the realist and the neoliberal theories. Many theories have used empirical data or formal deductions to prove their propositions, but there have been critical weaknesses in each approach. In the case of the empirical approach, the lack of enough real-world data has been a chronic disease even though we have improved our data-collection procedures and treatment methods. The formal approach has revealed a very strong way of theory-building with its deductive power, but it still is very slow in tracking real-world phenomena in a tangible manner. The distance between the data-driven empiricism and the theory-driven formalism is too far from each other to be integrated in a reasonable way. This thesis, considering this methodological gap in the both mainstream methods in international relations studies, selects the computer simulation method as a “halfway-house” (Holland, 1995). Although this method seems not to have been so popularly used in major subfields of international relations studies, many have noticed its efficiency as a secondary tool for theory-building or theory-testing. Detailed discussion about the way simulation methods are designed and conducted will be done in Chapter 3.

1.2 Puzzles

Based on these real-world motivations and theoretical contexts, this thesis tries to answer some theoretical puzzles related to understanding the conflict and cooperation in world politics. The puzzles concern the conceptual inquiry into the structural dynamics of
world politics, but they need to be developed into a more answerable one. Before I
describe the puzzles of this study in detail, I would like to discuss some points relevant to
them in the theoretical contexts. First, the puzzles try to capture major aspects of
international politics by comparing the internal mechanisms of conflict and cooperation,
and their consequent impact on world politics envisioned by the realist and the liberal
theories. In the model, conflict will be represented by war and cooperation by trade as
other major international relations models. Although the puzzle is more theory-driven
than real-data-driven, it can also make sense in many real-world policy contexts. In
particular, this thesis would try to deal with the general appropriateness of contrasting
policy recommendations suggested by major international relations theories, such as
power politics and cooperative strategies. While there are many factors that affect the
ultimate consequences of world politics, this thesis focuses on the “essential factors” –
such as power distribution, power estimation error, war cost ratio, and metabolism ratio
for trade – suggested by major theories of realism and liberalism. By clarifying essential
mechanisms from major theories of world politics, this thesis tries to estimate the features
and differences of them and figure out how they can be integrated into one.

Second, the puzzles consider the dynamic, not the static, mechanism of world
politics. The structure of world politics is always fluctuating across a lot of conditions –
conflict and cooperation, dominance and alliance, bipolarity and multipolarity, etc. –
defined by many contrasting theories. Although there are some features described as
embedded or fixed in a system, it is reasonable for us to assume the dynamic nature of
world politics in a longer, broader perspective. The dynamics of world politics is
understood as an observable phenomenon emerging out of the working of its fundamental
mechanisms, no matter what theoretical bases it depends on. For example, whether our model is founded on the realist theory, the liberal theory, or a mixed theory, it is intended to explain the macro-level phenomena like war, trade, alliance, and polarity. Theory supplies the micro-level elements for the explanatory mechanism, and this mechanism is used for explaining the macro-level phenomena. However, these macro-level phenomena always change their conditions so that the focus of this study is on what mechanisms cause the system to be dynamic. As such, this research intends to observe the systemic phenomena—structural dynamics—of world politics while explaining it by their fundamental, lower-level mechanisms. It sees world politics not as fixed at a certain point, but as moving across various conditions. Instead of asking why the world is in a specific condition, this thesis seeks to examine how world politics tends to reach a condition rather than another and under what conditions it does so.

Third, the puzzles also seek the end situation of world politics. It may be nonsense to directly ask about this, but the answer based on theoretical underpinnings might help us clarify the complicated connections between contradicting theoretical arguments. In this study, I will set up a virtual world of realism. In this virtual world, agents (mostly states) follow the realist rules of behavior. One probable expectation is that the world has a very high probability of collapsing into an empire, after a long period of brutal wars upon the rule of power politics. Perhaps we might expect a more cooperative and more stable world if we adapt a liberal theory to our description of world politics. Then, what would be the result of world politics in which both rules are taken by agents? Is it possible for us to anticipate any noticeable patterns out of different combinations of contradicting theoretical bases? If we do not rely on each extreme side, then what
proportion of each element (behavioral rules suggested by theories) should be accepted? Even with many different starting conditions with various parameters, we may expect something different at the end of each theoretical scenario. Perhaps we may reach an empire with some probability or a multi-state world with a different probability, if we let the artificial worlds run many times on the computer screen. We can collect a great deal of empirical data from the dynamic runs of our virtual worlds that are artificial but based upon existing international relations theories. Although we may not pinpoint the future scenarios of world politics, this kind of statistical work based on the notion of probability may have deep policy-relevant implications as well as theoretical ones.

Fourth, the puzzles proposed here imply the directionality of causal relations among major factors that have been proposed by major theories. In many theories, explanation connects micro-level factors and macro-level factors in a bottom-up or a top-down ways, so that they stress different aspects of the target phenomenon. The bottom-up approach starts from the lower-level factors such as individual characteristics, while the top-down one traces from the systemic, structural factors downward. This thesis focuses on the bottom-up approach, even though it does not ignore the top-down style. Motivations in the big puzzle imply the micro-level foundations of the consequential systemic phenomena observed at the macro-level. Considering any possibility of the reverse causal relationship between major factors in international relations theories, this research will also look into the structural factors. Particularly, the structural approach of neorealism will be contrasted to the conventional realism and liberalism that are based more on the individual-level rationality.
Finally, the puzzles presented here aims to emphasize the essential point of this study – the *level of analysis*. First of all, this research investigates the impact of unit-level (or agent-level) factors such as state option on the systemic phenomena such as polarity or balance of power patterns. At the same time, this study is also interested in the role of dyadic factors such as interdependence and dyad types between two interacting states on the systemic phenomena. Dyadic factors are especially important in this study as they are a feature of “interactions” among units rather than a feature of “units” themselves. Also, this study looks at the system-level factors, such as polarity and power concentration, which might be considered as the feedback effects on the unit-level. These factors are considered for a theoretical purpose of comparison, even though this study puts emphasis more on the micro-level factors (and their interactions) than on the macro-level ones. Upon the theoretical context discussed so far, the three major puzzles of this study are as follows:

**Puzzle 1:** How do the micro-level properties of state agents affect the macro-level structures of world politics?

**Puzzle 2:** How do the dyadic properties between state agents affect the probability of war in world politics?

**Puzzle 3:** How do the systemic properties of world politics affect the probability of war in world politics?
The first puzzle deals with the impact of individual agents' choices (the micro-level property) on the structure of world politics at the macro-level by setting the direction of causal relationship as "bottom-up." This puzzle is based on the basic assumption of many international relations theories – especially classical realism and liberalism – that the properties of individual agents affect the system of world politics. The major targets of investigation are a couple of structural dimensions of world politics – system multiplicity (or endurance), balance-of-power, and state survival rate – which may be affected by the change of state properties in modeling. As such, this puzzle assumes the causal impact of the micro-level factors on the part of state agent on those macro-level factors. This puzzle is also targeting the previous computer simulation works in international relations by adding the liberal, cooperative factors to the conventional model of world politics. This puzzle intends to draw the audiences from the classical realists, the neoliberal institutionalists, and the computational modeler in international relations.

The second puzzle examines the relationship between the dyadic properties of interacting agents and the consequent probability of war in the system. Here, I narrowed the target to the probability of war in the system in order to compare the results with existing theories. Considering that many empirical findings have shown that economic interdependence has negative impact on the probability of war, I tried to test it in my simulated world by proposing this puzzle. Also I want to find any meaningful conditions for the impact of the heterogeneity in dyads. Many international relations studies, especially those on the relationship between economic interdependence and conflict, have relied on the dyadic data. However, few have concentrated on the different nature of
dyadic relations such as the level of interdependence and the power difference between
the two countries on the same dyad. I would like to explore the possible range of
theoretical extension from the current debate by generating many different situations
regarding the different levels of economic interdependence and power between the two
states on the dyad. As such, this puzzle focuses on the properties of dyadic factors that
might have impact on the probability of war in the artificial system of states.

The third puzzle asks about the nature of the structural factors on the conflict
pattern in world politics. By presenting this puzzle, I would like to focus on the role of
system-level factors, such as polarity and power concentration, in war among states. This
point is closely related to the existing debate on the agent-structure relations. One of the
most important features of the simulation research design of this study is the “bottom-up”
approach, in which the structural target phenomena, such as polarity, balance-of-power,
power concentration, and system endurance are understood as emerging out of the
interactions among state agents. Therefore, I would like test how the structural factors
generated “bottom-up” – represented by polarity and the power concentration among
major states – affect the behavior of states, especially the probability of war. Detailed
propositions in a testable form will be established in Chapter 2 with their theoretical
implications. Figure 1.1 shows the overall structure of these puzzles.
<table>
<thead>
<tr>
<th><strong>Puzzles</strong></th>
<th><strong>Causes</strong></th>
<th><strong>Effects</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Puzzle 1</strong>&lt;br&gt;How do the micro-level properties of state agents affect the macro-level structures of world politics?</td>
<td>Unit-Level Factors (Agents' Properties)</td>
<td>Macro-Level Structures</td>
</tr>
<tr>
<td><strong>Puzzle 2</strong>&lt;br&gt;How do the dyadic properties between state agents affect the probability of war in world politics?</td>
<td>Dyad-Level Factors (Agents' Relationship)</td>
<td>Probability of War</td>
</tr>
<tr>
<td><strong>Puzzle 3</strong>&lt;br&gt;How do the systemic properties affect the probability of war in world politics?</td>
<td>System-Level Factors (Structures)</td>
<td>Probability of War</td>
</tr>
</tbody>
</table>

*Figure 1.1: Structure of Puzzles*
1.3 Description of Chapters

Chapter 2 reviews existing literature on the topics that are discussed in the previous sections. The tradition of trade and war studies, economic interdependence, and structural neorealism will be discussed concerning their implications on the major puzzles presented above and the propositions to be tested in this study. In addition to the survey of major theories and empirical studies relevant to major topics of this thesis, the tradition of cellular automata simulations in international relations will be summarized in terms of a prototypical method. Also theoretical propositions, based on the major puzzles suggested above, will be developed in a form to be tested with simulation methods. They will be reorganized into three groups according to their relationships to major puzzles, which will be tested in separate chapters. Research methods will be discussed in detail in Chapter 3, with detailed introduction to cellular automata simulations, object-oriented programming, agent-based modeling and major algorithms used in this study.

Chapter 4 discusses the impact of the micro-level properties of state agents, especially the addition of the trade option, on the structural patterns of world politics. Simulations will be designed and implemented to reflect the theoretical differences between classical realism and liberalism. The focus will be on the role of war and trade as the symbols of conflict and cooperation between states in world politics. Two artificial worlds – the security-only world and the security-and-trade world – will be built separately in order to test the propositions regarding state choices. Using these separate sets, this study will investigate whether and how the choice of states between war and trade, based on the rationality assumption, changes the structure of world politics at the
system level, in contrast to the classical realist, security-only world. Three dependent variables – system endurance, balance of power, and state survival – will be traced at the system-level upon different combinations of major factors, especially with and without state choice of trade. The simulation framework with the trade module, as an alternate simulation model to the conventional security-only framework, will be used as a base model for the theoretical extensions in the next chapters.

Chapter 5 deals with the role of dyadic properties on the probability of war in world politics. Considering that the topic – the relationship between interdependence (and the types of dyads) and conflict – has recently attracted many scholars’ interest, this thesis will run simulations that adapt this theme to artificial worlds. The target of the test is narrowed down from the previous chapters – i.e., from the three systemic properties to the probability of war – because of the necessity of close connection to existing theories and empirical findings. Also, while the previous chapter has a dual mission – examining the difference between the security-only world and the security-and-trade world, and establishing a prototype of a combined simulation model – this chapter is an extension of the latter mission into one of hottest topics in international relations debates.

Chapter 6 investigates the influence of systemic factors on the probability of war. As briefly mentioned before, this chapter is a test for the structural neorealist theories about the conflict dynamics of world politics. Also it is a kind of self-test for the bottom-up approach of this study as it contrasts with the most powerful theory in international relations, the structural neorealism. I would like to explore the roles of structural factors on the behavior of state agents in my simulation and integrate these to my original
bottom-up theory. Polarity and power concentration will be used as the main structural factors to test here.

The final chapter will wrap up the findings of this study and discuss their implications on existing international relations theories. Regarding real-world politics, some policy implications of the findings will be discussed, too. In addition, some comments on the limitations in this simulation study will be added because this uses a quite new technique which is not yet familiar to many political scientists. Based upon the limitations discussed, a research agenda for the future will be suggested to further the findings of this study.
CHAPTER 2

THEORY AND PROPOSITIONS

2.1 Theoretical Contexts and Literature Review

This research is based on a critical review of several research traditions in international relations – war and trade in classical realism and liberalism, interdependence and peace studies, structural neorealism, cellular automata computer simulations, and the newly emerging paradigm of complex systems theory. The issue of conflict-cooperation relations among state agents has been one of the most controversial issues among political practitioners as well as among international relations theorists. In this chapter, I will discuss what kind of theoretical traditions this research follows and what propositions can be deduced from them, regarding the three major puzzles discussed in the previous chapter. Regarding the main topic of this research, I would discuss war and trade as “state choices,” upon the realist-liberal assumption of individual rationality. Recent theories about the relationship between interdependence and conflict will be introduced, too, with their limitations in empirical and theoretical dimensions. The Waltzian tradition of structural realism will also be discussed as it represents a
contrasting approach to the theoretical framework of this study. Also a brief summary of
the tradition of cellular automata computer simulations in international relations will be
presented, with their implications for existing theories and this thesis. Finally, I will
mention the complex systems theory whenever appropriate to the main theme of this
study, even though it does not take a whole section in this chapter.

2.1.1 Trade and War as State Choices

The state, conceived as a unitary actor in world politics, has been so popular in
political science that it has been adopted as an assumption in many international relations
theories. Although various kinds of agent (or actor) have emerged in international
relations, such as transnational organizations, international regimes, and multinational
corporations, the role of state as a major unit of analysis seems not have decreased at all.
The state, in the major traditions of international relations studies, has been modeled as
holding policy initiatives upon their own sovereignty. In other words, we understand that
they “choose” their own behaviors in interacting with other states (Jervis, 1997, 4).
Sometimes states are at odds with neighbor states, while cooperating with them at other
times. Whatever reasons and motivations are hidden behind their behaviors, they choose
what they want to do by themselves upon the assumption that they have “power” to do so.
Power has been regarded both as an ultimate goal and as a means of state agents in world
politics. The state, in many international relations theories, has been assumed as setting
its goal at “maximizing power” and/or “maximizing welfare” by all available means
(Hirschman, 1948, 4). Although there have been unending disputes about its definition
and operationalization, the notion of "state as unitary actor maximizing its power/wealth" is still one of the most dependable assumptions in modeling international relations. This study takes the same assumption of the state as a rational power/wealth maximizer in developing theoretical propositions.

Choice is essential for a rational state as far as its decision makers choose strategies from a set of multiple alternatives. States are not always constrained by the structure of international system to conduct only war or conflict. If we focus on the internal "process" rather than the external "structure" of world politics, then we can easily acknowledge the importance of state choices (Russett, 1995, 268).1 States choose their behaviors toward others in the anarchic world, and war is one of the most favorite choices for states throughout history. In the realist tradition, as such, war and conflict have been the most important research topics in studying world politics. Individual states, according to the realist theory, follow the Hobbesian rule of primitive power seeking - the human struggle against all other humans - based on their own power. On the other hand, liberals have put their emphasis on the cooperative dimension of states' interactions. Trade, along with other kinds of inter-state relations, has been regarded as the most symbolic form of cooperative behavior among states. The liberals' idea of cooperation and trade has opened an alternative or complementary way of modeling international relations vis-à-vis the conventional realist framework.

As such, in the major traditions of international relations theories, war (as conflict) and trade (as cooperation) have been the two symbolic behaviors chosen by

---

1 Particularly, Russett focused on the link between domestic politics and international strategies when he talked about the dimension of "choice" in state decisions. I would not consider domestic factors in this study, even though I agree with his bottom-up framework.
states, rather than determined by environments or system structures. While there are many other observable phenomena between interacting states, these two "selectable paths" have been modeled as two major forms of behaviors in world politics. Each path has been understood as a means to support the other and as a goal to be sought by the other. Actually, the mercantilist idea of state power for military force combined by economic wealth has been a popular starting point in building an international relations theory (Viner, 1948). Albert O. Hirschman also proposed the notion of foreign trade as an instrument of national power politics and the political effect of economic behavior. Trade, according to him, is an "attribute of national sovereignty" (1948, 12-15). For decades since these early theories on political economy, the dynamics of state power and war-and-trade dualism has been a popular topic in the study of international relations.

The discussion of state choice between war and trade is closely linked to the proposition building in the next section, because the comparison of the security-only model and the security-and-trade model in international relations theories is one of the major purposes of this study. While accepting the assumption of the state as a rational actor, this study ponders whether there is any difference between the classical realism and liberalism at the (observable) system-level. We know that the world is not always war-ridden, and that states know how to cooperate with each other in many cases. Considering this, Rosecrance (1986) developed a liberal theme about the effects of economic interdependence and the consequent emergence of "trading state" in world politics. His basic argument is that the world can escape from the vicious cycle of conflicts with the growing interdependence among countries. The shift toward an interdependent trading world from the old military-political-territorial world has been accelerated since World
War II, as the risk of war (especially nuclear war) increased and the economic interdependence between states increased (1986, 13-14). He saw that war (i.e., territorial expansion) or trade is a matter of “choice” for states, under the assumption of functional division of labor and reciprocal exchanges between them. That is, war and trade are regarded as selectable options or policies for a state toward other states. Clearly we may get an interesting comparison between the classical realists’ world (the military-political-territorial world in Rosecrance’s term) and the liberal world (the trading world) following Rosecrance’s propositions.

Although we understand that the factor of cooperation such as trade is important in understanding and explaining world politics, how much and by what mechanisms it is so are hard to answer with pure theory or with the scarce empirical data. In this sense, the experiment of this study using computer simulation methods can contribute to the explication of the hot theoretical issue – how much is cooperation important in modeling world politics? – between theories. Also this study would compare the effects of unit-level attributes of state agents (in the dual setting with/without the trade option) on the structure of world politics. This might be a challenging project as the result can reveal whether the classical realists’ and the liberals’ scenarios end with the same consequence or not, at the systemic level. In addition to this mission, the model for the security-and-trade world based upon the rationality assumption (and state choices) will be used as a base model for the extension of theoretical tests in Chapter 5 and Chapter 6.

---

2 Rosecrance uses the term “dualism” in order to emphasize his idea of war and trade as the selectable state decisions. He argues that the path to war or trade is finally determined by states themselves, rather than predetermined by structure. In this context, he distinguishes his theory from another dualism by Keohane and Nye (1977) who see interdependence as a systemic attribute (Rosecrance, 1986, 62-63).
2.1.2 Interdependence and War

Since the end of World War II, increasing interdependence between states' economies—such as the removal of trade restrictions, reduced transportation costs, and balance of payment institutions—and its impact on world politics have drawn political scientists' attention. Interestingly, a strong initiative in this direction was taken by an economist. Richard Cooper (1968) studied the phenomenon of increasing economic interdependence in the post-War period as a restraint on state autonomy. According to him, the growing level of interdependence makes each state's policy difficult to implement and even worsens interacting countries' domestic economies (Cooper, 1968, 148). The uncontrollable policy framework, in which states are inter-connected, was the main topic of his discussion on interdependence, even though the mutual exchange of resources across state boundaries is regarded as increasing the participants' welfare level in the economic perspective. He proposed what many political scientists should have done before him—a strong negative feedback of complex economic interdependence on politics. States have caused themselves to be interconnected with each other in order to increase their welfare level; but, as a mirror image of the same behavior, they have become constrained by those interconnections.

When Cooper investigated the relationship between Western Europe and America, his basic idea was that growing international economic interdependence reflects and causes a much higher sensitivity of transactions to each side's economic condition (1968, 59). He followed the classical economic theory of comparative advantage that focuses on the mutual benefits of trade relationship. That is, different countries with different levels...
of natural endowments or production resources generate cost differences between themselves, so that the potential structure of production, in which states are interconnected, will be established at the systemic level. With the convergence of these production cost structures, Cooper argued, the sensitivity of trade has become larger even with relatively small changes in each other's economy (1968, 76).

Since the theoretical contribution by Cooper, many political scientists have engaged themselves in the discourse of economic interdependence – such as Rosecrance and Stein (1973), Keohane and Nye (1977) and Waltz (1970 and 1979). While this study does not intend to cover the whole range of theoretical debates on this topic, some important empirical works regarding the influence of economic interdependence are worthy of mentioning. Although there have been a lot of suggestions on conceptualizing and measuring interdependence in international relations studies, we have yet to observe any theoretical agreement on them. Nevertheless, empirical works have converged on some common patterns and prototypes in these matters, particularly with regard to the economic aspect of interdependence which is the second theme of this thesis – How much economic interdependence at the dyadic level affects world politics at the systemic level.

Solomon Polachek (1980) was one of the first to systematically investigated the relationship between conflict and trade. He measured the level of interdependence by trade volume, and found support for his proposition that dyadic conflicts are negatively related to dyadic interdependence patterns. According to his empirical findings, a doubling of trade between two countries lessens hostility between them by twenty percent. One important implication of his findings is, as he mentioned, that peace cannot be
achieved just by imposition but by setting interdependent (i.e., trade) relationship between two countries. This is closely related to the main theme of the complex systems framework that a cooperative order can be self-organized from the Hobbesian chaos of world politics. Also, another implication of his findings is that conflict is more sensitive to trade, like Cooper's arguments (1968), than many economists think. More trade means more interdependence, and this leads to a stronger restriction of war and conflict.

Polachek continued his research with his associates on the same topic—especially on the impact of trade (as a symbol of interdependence) on conflict. Gasiorowski and Polachek (1982) focused on the East-West trade linkages in the détente period and found the same pattern that the growth of trade had caused a substantial reduction of U.S.-Warsaw Pact conflicts. This finding is closely related to the pattern of complex interdependence between superpowers and their bloc members in that it reduces the level of conflict by providing mutual benefits to partners in the way called issue-linkage (Gasiorowski and Polachek, 1982, 710-711). Polachek and associates, in other research (Polachek, Robst, and Chang, 1999), extended the original model of trade and conflict beyond the dyadic relationship by including third party factors and other issues such as contiguity, foreign aid, tariffs, and state size. The overall result from their study, despite some variations across variables, still supports the classic proposition that trade reduces the level of conflict.

Domke's work (1988) also supported the liberal theme that trade decreases war frequency, even though he used state as a basic unit of analysis rather than interstate dyads. A couple of studies by Oneal and Russett (1997 and 1999a) also have followed the same routes in their empirical investigation of interdependence. Their works mostly...
measured the impact of democratization with other control factors, in which they showed a negative relationship between interdependence and conflict. Oneal and his associates (1996) also supported the liberal thesis of interdependence and war in the context of democratic peace theory. A recent study by Gartzke, Li and Boehmer (2001) explored the interdependence-conflict relationship in more detailed areas such as joint currency and capital investment, beyond the classic sector of trade. However, these studies measured the impact of interdependence or trade with many other control variables - like democracy, economic growth, geographic contiguity, alliances, and power ratio. Thus the study of interdependence-conflict relationship has been extended from the simple trade-war dichotomy to the complicated multi-dimensional areas of international relations.

Interestingly, there have been some contrasting results in empirical studies that may harm the empirical support of the liberal theme. For example, Katherine Barbieri (1996) found a mixed result in the influence of interdependence – that is, she discovered a pattern of curvilinear relationships between interdependence and conflict. In a response to Barbieri, Oneal and Russett (1999c) also found some mixed results about the relationship between interdependence and conflict, especially with the introduction of a power difference variable, even though they supported their old liberal thesis. Lois W. Sayrs (1989) also proposed a mixed-style thesis that trade may increase or decrease cooperation/conflict levels. Table 2.1 is a summary of major empirical works on the topic of interdependence and war.
<table>
<thead>
<tr>
<th>Authors*</th>
<th>Obs. Type</th>
<th>Obs. Period</th>
<th>Obs. Numbers</th>
<th>Dependent Variable</th>
<th>DV Data Source</th>
<th>Independent Variable</th>
<th>IV Data Source</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polachek</td>
<td>State dyads</td>
<td>1958-1967</td>
<td>N = 9,000</td>
<td>Net conflict frequency = Conflict frequency - Cooperation frequency</td>
<td>COPDAB</td>
<td>Trade = Imports and exports (dollar values)</td>
<td>Gillespie &amp; Zinnes Banks</td>
<td>Liberal</td>
</tr>
<tr>
<td>(1980, JCR)</td>
<td></td>
<td></td>
<td>(All possible dyads N = 126,150 of 30 states)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1986, ISQ)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Vulnerability interdependence = Export commodity concentration Capital flows</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polachek</td>
<td>State dyads</td>
<td>1967-1978</td>
<td>US-Warsaw States (7) N = 130</td>
<td>Net conflict frequency</td>
<td>COPDAB</td>
<td>Trade = Imports and exports (dollar values)</td>
<td>IMF US Dep't of Commerce</td>
<td>Liberal</td>
</tr>
<tr>
<td>(1982, JCR)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polachek, Robst</td>
<td>State dyads</td>
<td>1958-1967</td>
<td>30 states</td>
<td>Net conflict frequency</td>
<td>COPDAB</td>
<td>Trade (volumes)</td>
<td>IMF etc.</td>
<td>Liberal</td>
</tr>
<tr>
<td>Chang (1999, JPR)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Control variables: Country attributes, Tariffs, Foreign aid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polachek, Oneal</td>
<td>State dyads</td>
<td>1950-1985</td>
<td>Politically relevant pairs N = 6,600 ~ 7,200</td>
<td>Militarized disputes (dichotomous)</td>
<td>COW</td>
<td>Interdependence = (Export + Import) / GDP</td>
<td>IMF</td>
<td>Liberal</td>
</tr>
<tr>
<td>Maoz, Russell</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Control variables: Democracy, Institutional constraints, non-violent norms, Alliances, Contiguity, Military capability ratio, Economic growth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1996, JPR)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Continued)</td>
</tr>
</tbody>
</table>

Table 2.1: Empirical Studies on Interdependence and Conflict
<table>
<thead>
<tr>
<th>Author</th>
<th>Year</th>
<th>Type</th>
<th>Time Period</th>
<th>Sample Size</th>
<th>Militarized Disputes</th>
<th>Interdependence (dyadic)</th>
<th>Control Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>dyad states</td>
<td></td>
<td>War N = 213</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>disputes</td>
<td></td>
<td>Interdependence = Salience x Symmetry, Control variables: Joint democracy, Alliances, Capability ratio</td>
</tr>
<tr>
<td>Oneal Russo</td>
<td>1999, WP</td>
<td>State dyads</td>
<td>1885-1992</td>
<td>N = 15,000</td>
<td>Militarized</td>
<td>COW</td>
<td>Interdependence = (Export + Import) / GDP, Control variables: Democracy, Joint IGO, Capability ratio, Alliances, Contiguity, Kantian &amp; Realist variables</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Excluding</td>
<td></td>
<td>disputes</td>
<td>MID</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1915-1920,</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1940-1946)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>disputes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Barbieri</td>
<td>1996, JPR</td>
<td>State dyads</td>
<td>1879-1938</td>
<td>N = 14,341</td>
<td>Conflict</td>
<td>COW</td>
<td>Interdependence salience = (TS_L x TS_H) / 2, where TS_L = Trade_L / Total trade,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Interdependence symmetry = 1 -</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Interdependence = Salience x Symmetry</td>
</tr>
<tr>
<td>Table 2.1: Continued</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Mansfield (1994)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System 1850-1964 5-year period</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>War - Major</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Non-major</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Singer &amp; Small (1972)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trade (systemic) = Global exports / Global production</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Singer &amp; Small (1972)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BdM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Levy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Levy (1972)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Trade (systemic) = Global exports / Global production</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Control variables:</strong> Hegemony, Polarity, Capability concentration (CON), Average CON change, Average capability share change, Openness of trading system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| **Sayrs (1989, II)** |
| State dyads 1950-1975 |
| N = 4,472 |
| COPDAB |
| Conflict & Cooperation (Separate models) |
| **Trade** |
| = ((Dyadic export / Actor GDP) + (Dyadic import / Actor GDP) + (Dyadic export / World export) + (Dyadic import / World import)) / 4 |
| **Control variables:** Reciprocity, Memory |

| **Gartzke Li & Boehmer (2001, IO)** |
| State dyads 1951-1985 |
| Contiguous major powers N = 622 |
| Monetary: 1966-1985 |
| Monetary N = 10,399 |
| Capital N = 2,200 |
| Conflict (Threats, disputes, use of force, war: dichotomous) |
| MID |
| **Monetary interdependence** |
| - Pegging |
| - Joint currency area |
| Control variables: Regime type, Trade interdependence, Contiguity, Allies, Capability ratio |

| **Domke (1988)** |
| State units 1815-1986 |
| War: 1871-1975 |
| N = 217 |
| Decision to war (for a state) |
| Singer & Small (1972) |
| **Trade Ratio** |
| = Export / GNP |
| **World Bank** |
| **Liberal** |

Besides these studies, there are others that used similar sets of variables but that focused on the relationship between interdependence (or trade) and conflict (or politics) in a reversed causal direction. Brian M. Pollins (1989a, 1989b) showed that trade flows are influenced by political arrangements between countries. Barbieri and Levy (1999) also measured the impact of war on trade both in the short term and the long term. While the main concern of this thesis is not this kind of war-to-trade impacts, the implication of this type of literature might be significant for future research agenda, especially for modeling the dynamics of the relationship. Actually, a couple of scholars have tried to overcome the conventional liberal frame by incorporating both war (politics) and trade (interdependence) into the same equations, in order to see which variable has stronger impacts than others. Polachek (1980), Gasiorowski and Polachek (1982), and Reuveney and Kang (1996 and 1998) are examples of this. Although political scientists have confirmed that trade and interdependence have great impacts on war and conflict patterns, they might have to turn their attention to the mixed results and the reverse causal relationship as well.

The “theory” of interdependence, war and trade seems to have been established in the study of international relations these days in this way. Empirical evidence, however, seems to have left more room to back up the theory. Here I will discuss some weak points regarding the limits of existing empirical studies before I discuss why I would like to tackle this theme in this thesis. First of all, one of the severe problems of existing interdependence-conflict studies is the lack of cases. Many empirical studies, due to the meager number of cases, fall short of self-sufficiency for verifying theoretical themes regarding interdependence and its impacts on conflicts. For example, Polachek’s original
study (1980) and a recent one with his associates (Polachek, Robst, and Chang, 1999) were based on the total 9,000 dyads of conflict from 126,150 combinations of 30 countries that were extracted from COPDAB. This covers only the 10-year period between 1958 and 1967. This period had been severely influenced by the Cold War and the nuclear factor. The selected number of thirty that were used in explaining frequent conflicts in the Third World area might cause a selection bias. Another study by Gasiorowski and Polachek (1982) was more restricted in its coverage of cases with only 130 cases of conflict between only seven countries (the U.S. and the Warsaw member states). While Polachek and his associates' studies were pioneering works in this field, they might not always be generalized due to this small-N problem.

There are many other studies that chose the cases only from the latter part of twentieth century. Sayrs (1989) worked with 4,472 cases between 1950 and 1975, so that he could not overcome the same problem of small-N that afflicted Polachek and others, either. Oneal and Russett (1997 and 1999) conducted their investigations on the cases extracted from 1950-1985 and 1950-1992, respectively. The first study examined the conflict relationship (N = 213) only between contiguous major powers, while the second looked at the war cases (N = 166) from all dyads. A recent study by Gartzke and associates (2001) also set the limits of time span between 1951 and 1985 for conflict and between 1966 and 1985 for monetary interdependence. On the other hand, Barbieri (1996) used only 14 cases of wars and 270 cases of military disputes from the 14,341 dyads during 1879 and 1938 so that ignored the post-War cases. As such, these studies

---

3 Another of Gasiorowski’s studies (1986) expanded the time span, but it still covered less than 20 years between 1960 and 1977.
4 Oneal, Oneal, Maoz, and Russett’s work (1996) also focused on the 1950-1985 period, with about 7,000 “politically relevant” cases.
might be prone to selection bias by ignoring many other cases in the periods that might not have been under the same structure and dynamics as those of the post- or pre-World War II.

Nevertheless, due to the development of data manipulation techniques, the range of time span in the study of war and interdependence has been dramatically expanded up to the early 1800’s. For instance, Domke’s study (1988) was based on the 217 war cases from the period between 1815 and 1986 (with the trade relationship between 1871 and 1975). Mansfield’s work (1994) also extended the range of war cases between 1850 and 1964. Oneal and Russett, in one of their recent studies (1999), also covered the time period between 1885 and 1992, while they excluded the two World War periods. This enabled them to use more cases – 15,000 dyads and 6,000 military disputes. This seems a desirable trend in the study of war and interdependence, but we still have many problems in the collection of relevant data and their manipulation. Perhaps we cannot expand the range of our studies of conflict and trade anymore beyond the nineteenth century due to the lack of abstract historical records. Although it may not be impossible, the lack of data necessitates more labor in coordinating datasets to be compatible with each other.

Another trend to note in empirical studies is that the property of study is strongly dependent on the nature of original dataset. Most of Polachek and his associates’ studies have been using COPDAB for conflict variable that were coded as dyads from the start. However, the level of conflict and cooperation in COPDAB was coded according to an arbitrary scaling system. Also COPDAB does not cover the pre-World War period, so that it is severely biased toward the hegemonic or bipolar periods. On the other hand, many of Oneal and Russett’s studies rely on the Correlates of War (COW) dataset.
Sometimes people added information from the MIDs, like Oneal and Russett (1999) and Gartzke et al (2001). Recently the range of COW has expanded to the mid-1800's. In the case of trade data, the IMF and World Bank have been two major sources of trade-related data. In addition to these, personally compiled datasets were used in some studies, like Bueno de Mesquita's and Levy's war data (Mansfield, 1994) and Gillespie and Zinnes' (Polachek, 1980) Angus and Maddison's (Oneal and Russett, 1999), Barbieri's (Barbieri, 1996), Kuczynski's (Mansfield, 1994) trade datasets. As such, the studies on war and interdependence could not avoid being influenced by the very nature of the original datasets that were used.

The reference in the selection of cases for each study should be mentioned, too. Theoretically, there might be many cases of dyadic interactions even during a short time span. However, many studies have focused only on major power relations while ignoring small actors. This may put a serious restriction on the implications of those studies. That is, the theory of interdependence and war might be a theory of major powers that cannot be applied to small ones and on the relations between great powers and small ones. This problem seems to be severe in Oneal and Russett's studies. For example, in one of their studies (1996), they chose the cases of dyads that are "politically relevant" that are "contiguous and containing at least one state defined as a major power by the Correlates of War project" (1996, 14). They intentionally excluded the cases that did not have a "reasonable opportunity to engage in armed conflict because they were too far apart, too weak militarily, and had few serious interests at stake" (Ibid.). This assumes that the structure and dynamics of world politics is dependent only on major powers' interactions. In many contexts, we need not accept this kind of assumption in its own right. No
selection bias should be the base of our study, even though it is not a technically driven motivation during the process of data analysis.

The measurement of core concepts is another big problem for the study of interdependence and war studies. First of all, the concept of interdependence seems to have held a couple of meanings for scholars since its first usage, even at the dyadic level. Following the non-standardized usage of interdependence, empirical works also have divided themselves in their operationalization of interdependence. Polachek (1980), Gasiorowski and Polachek (1982), and Gasiorowski et al (1999) measured the level of interdependence by the total volume of trade. They assume that the level of interdependence between two countries increases if the combined level of trade (import and export for one country) grows. On the other hand, Gasiorowski (1986) detailed his measurement by dividing the concept into two — sensitivity and vulnerability. The former, according to him, can be measured by import price elasticity and the latter by export commodity concentration. He combined these indices with capital flows to measure the level of interdependence. However, it is not clear whether this measurement conforms to the theoretical sophistication of sensitivity and vulnerability of interdependence, suggested by Keohane and Nye (1977) and Waltz (1979).

Other scholars have been more prudent in measuring interdependence. In particular, some denied the measurement of interdependence level by the pure volume of trade. For example, Oneal and Russett (1997, 1999a, 1999b), Oneal and associates (1996) gave weight to the trade volume by dividing export plus import by each state's GDP. Domke (1988) also measured interdependence by the export volume divided by GNP level. Mansfield (1994) used the index of export to the total production ratio, even if it
measures the world total rather than the dyadic one. Other scholars gave more weight to the size variable in calculating interdependence levels. For example, Sayrs (1989) measured interdependence by trade weighted both by individual actor’s import/export and world import/export level. One of the most sophisticated indices was made by Barbieri (1996). She measured interdependence by the combined level of salience and symmetry – “salience” indicating the product of each country’s ratio of trade toward the other to the total trade, while “symmetry” showing the level of difference between two states’ trade levels toward each other. As such, the measurement of interdependence has been developed and elaborated with more consideration of important factors, but there seems not to be any standard indicator of the level of interdependence.

The measurement of conflict and war also has not shown any agreement between scholars. Many followed the original raw data in measuring the happening of conflict/war in dyadic years. For example, Polachek (1980) measured the net conflict frequency by subtracting conflict frequency from cooperation frequency. Sayrs (1989) also measured the level of conflict and cooperation in a similar manner, following the 15-scale criterion of COPDAB. While Polachek and his associates (Gasiorowski and Polachek, 1982; Polachek, Robst, and Chang, 1999) calculated the ordinal level of conflict, other scholars who used COW dataset had to code the incidence of militarized conflict into a binary variable. Oneal and associates (1996) and Oneal and Russett (1997, 1999a, 1999b, 1999c) used a dichotomous variable of whether a militarized conflict happened or not for their studies. In the case of Domke (1988), because of his unit of analysis – state units rather than state dyads, each state’s war decision was counted as a conflict measure. Only Mansfield (1994) measured the case of war (both major and non-major) in his study by

32
combining both COW and other datasets such as Bueno de Mesquita's and Levy's. Other people expanded the measurement of conflict by including threats, several kinds of disputes, use of force as well as normal wars in their calculation (Barbieri, 1996; Gartzke, Li, and Boehmer, 2001).

Considering the weaknesses revealed in existing empirical studies of interdependence, this thesis tries to complement them by the computer simulation method. As mentioned above, one of the severe problems in the study of interdependence is the shortage of relevant cases. While we study the impact of economic interdependence on political and military relations, we cover only a period of less than two hundred years at best, in which only a couple of structural patterns have emerged. Simulation studies can supplement these restrictions by generating many theoretically relevant situations on the computer screen. In particular, this thesis tries to test the proposition that economic interdependence at the dyadic level decreases the level of conflict at the systemic level, which has been a popular topic for the last decades regarding this topic, this time with artificial datasets that resemble world politics.

2.1.3 Neorealism, Constructivism and International Structure

One of the characteristics that distinguish the Waltzian neorealism from classical realism is its focus on structure as a major independent variable. According to Waltz, the behavioral patterns of world politics are influenced by systemic factors such as the change in the distribution of capabilities among states. He criticizes the reductionist tradition of existing theories in explaining the patterns of world politics by dissecting
micro-level factors smaller and smaller (Waltz, 1979, Chapter 4). His emphasis on structural factors, however, has been the target of many critiques, among which the most significant one has its roots in the critical, constructivist theory. The most compelling reason of this critique is the Waltzian treatment of structure as “given,” while constructivists have identified themselves by defining structure in an endogenous way. That is, they have argued that structures cannot generate themselves but are created out of the interactions of its members (Wendt, 1987, 1992 & 1994).

Unlike the conventional theories of neorealism and neoliberal institutionalism, the constructive theory treats actors and structures as codetermined and intersubjective. The behaviors of agents and the structure itself are understood in social contexts. In this sense, the constructivist theory regards anarchy as “imaginary” in the neorealist theory. In their perspective, anarchy is itself “constituted” by states themselves (Wendt, 1994). State identities also depend on context, so that states are assumed as not having a priori interests as in the conventional neorealist or neoliberal theories. Power is understood as discursive as well as material, in this perspective (Hopf. 1998. 172-181). In particular, regarding the role of structure, the constructivist theory critiques Waltz’s confusion of the reductionist paradigm. According to Wendt, Waltz was wrong in rejecting both the unit-level explanations and the interaction-level theories as reductionism. Interactions, in Wendt’s perspective, are another dimension of structure, which are mutually constructed among state agents.⁵ Also, according to him, Waltz “reified” structure just by separating it from agents and their interactions (Wendt. 1999. 145-147).

---

⁵ Wendt calls this level “micro-structure” from the agents’ point of view, while calling Waltz’s concept of structure “macro-structure” (Wendt. 1999. 146-147).
Considering the weight of these theories and debates between them in international relations, this thesis tries to test a proposition regarding the impact of structural factors. Both neorealism and constructivism regard structural factors important, but they are different in the treatment of structure: neorealism regards it as "given" while constructivism regards it as "constructive." This study resembles constructivism in its treatment of structure: the structural factors are not "given" but "constitutive" out of the interactions among state agents. The computer models used in this study do not assume systemic or structural properties but treat them as "emerging" out of the interaction-level among units. Therefore, one of the purposes of this study regarding the debate between neorealism and constructivism is to build a model of interstate interactions that generate the macro-level structures such as polarity (which Waltz and Wendt mentioned).

On the other hand, this study does not propose the same thesis of the "codetermination between structure and agents" like constructivism. Regarding this point, I would like to discuss the theoretical basis of this study in detail. The theory of complex adaptive systems (CAS) theory has recently been introduced to the social sciences which have been dominated by the Newtonian deterministic paradigm. As far as the epistemological perspective is concerned, the CAS approach regards the "complexity" of the target system we are investigating as constructed by the interaction among "individual agents" that are appropriate at the "unique" level of research. For example, physicists focus on atoms, biologists on cells, psychologists on people, sociologists on social actors, international relations theorists on state actors, and so on. At the same time, they wrap up their bottom-up theories with the observations at the macro-level phenomena that are the targets of their research. If the research stops here and tries to explain the macro-
phenomena with the micro-level factors at deeper and deeper levels, then it falls into the “trap of reductionism” as Waltz criticized. The bottom line is that the CAS theory adds the factor of “interactions” among these agents on top of this reductionist-like approach. Thus CAS-based social scientists study social phenomena, but they do not rely only on the properties of sub-level agents per se in explaining them. Instead, they focus more on the functional interactions among themselves that contribute to the generation of macro-level social phenomena. The same logic applies to the study of international relations – states are the major unit of analysis, but the macro-level phenomena are to be explained by the “interactions” of these individual states. We do not always need to get into domestic politics, into the psychology of decision-makers, or even into the biological cells in order to explain macro-level international relations phenomena. If any study identifies the major agent type (or dynamic unit) and theorize the mechanism of interactions among agents, then it needs not be trapped into the reductionist error. In this sense, the CAS approach should not be criticized as “reductionist.”

Interestingly, as mentioned before, the CAS theory has been regarded as sharing many similarities with the constructivist theory in their critique of the neorealist paradigm. Actually there have been some efforts to integrate the CAS approach and the constructivist paradigm (See Lyotard, 1989). They seem to be similar in their concentration on “interactions” among agents, rather than on agents per se or structures per se. However, the constructivist theory goes further in the development of an “interactionist” theory by putting the same weights on agent and structure. So neither “bottom-up” nor “top-down” can be properly dubbed the constructivist paradigm. It may be better to call it “bi-directional,” “up-and-down,” or “circulating” as the causal
relationship between factors is two-way. On the other hand, the CAS approach does not try to see factors in that redundant way; instead, many theoretical and empirical works in the CAS paradigm have been done only in the bottom-up framework. Due to the lack of relevant theoretical “starting points” – whether it is agent or structure or any other middle-level units – the constructivist approach seems to get bogged in the trap of circular logic in terms of theoretical and empirical aspects.

Anyhow the basic bottom-up approach – of course, with its focus on interactions – of the CAS approach is now clearly contrasted to other approaches. In particular, the top-down approach by the Waltzian structural realism seems to contradict the bottom-up style of this thesis. The bottom-up approach does not take the constructivist, circular logic in explaining the causal effects between structure and agents. Thus, in addition to the purpose of the previously mentioned modeling of the relationship between structure and agents without assuming the structural elements as “given,” this study also intends to compare and test the propositions regarding the impact of structural factors on state behaviors (especially war) in the bottom-up model of world politics. By this way, this study tries to investigate under what conditions structural factors influence on the probability of war. Thus this study will distinguish itself from the neorealist position in its treatment of interactions not as “reductionist.” It will also distinguish itself from the constructivist approach by starting from the “interaction-level” rather than being trapped in the circular logic between structure and agents. Therefore, this study intends to compare the structural neorealism and the constructivist approach in dealing with the structural factors in building a model of world politics.
2.1.4 Cellular Automata Simulations in International Relations

Another tradition to discuss in this section is the use of computer-based cellular automata simulations in international relations, which has not been very popular since its introduction to the social sciences decades ago. I would like to talk about it because the recent debate on interdependence and war has its roots in the old realist-liberal debate. Essentially, with regard to the research method of this study, testing existing theories with simulations deserve our attention. A great advantage of simulation studies, especially that use computers, is that they can explore any kind of dynamic system without restrictions. Many acknowledge that it helps and supplements real data analyses by clarifying theoretical working mechanisms and by experimenting with target events in a controllable manner. For this reason, simulation techniques have been used in international relations for decades and developed according to the evolution of computer technologies.

The tradition of computer simulations in international relations has its roots in the old Richardson model of an arms race in which wars are predicted from the arms race based on two differential equations.\(^6\) His model has provided two important prototypes for the current cellular automata simulations. First, it was a dynamic model so that it stimulated the study of the change in interstate relations. Secondly, it was composed of only two simple equations which are linear. This point is important in the context of the

---

\(^6\) Richardson’s arms race model between two nations is expressed with the following two equations: (1) \(\frac{dx}{dt} = ky - ax + g\), and (2) \(\frac{dy}{dt} = lx - fy + h\) where \(k\) and \(l\) are defense coefficients, \(a\) and \(f\) are fatigue coefficients, and \(g\) and \(h\) are grievances.
scope of simulations, because overly simple simulations may not be successful in predicting real-world phenomena (For a brief introduction of Richardson’s model, see Smoker, 1985). Nevertheless, Richardson seems to have implemented the principle of Occam’s Razor in his modeling so that it is a prototype of computer simulations which try to extract only the essential elements in a theoretical sense. The cellular automata computer simulation follows this philosophy of Richardson’s simple logic — “Do not make the model unnecessarily complicated.”

In this context, the later generation of world modeling has gone far from Richardson’s intention in that computer simulations have been used for global level problem diagnosis and prescriptions. The early effort of this kind, the Forrester-Meadows model started from the so-called “World Problematique” proposed by Club of Rome for a scope of world problems, even though it did not explicitly embrace war and conflict processes. Later simulations, like World Integrated Model (WIM) and the Bariloche Model, follow the original format of the Forrester-Meadows model with the same purpose despite some difference in their scopes. Harold Guetzkow’s Inter-Nation Simulation (INS) was the first simulation study from politically-oriented scholars, but computers were not used then (Ashley, 1983). Later, Stuart Bremer’s “Simulated International Processer” (SIPER) and the “Generating Long-term Options by Using Simulation” (GLOBUS) started as the mainstream computer simulations in international relations (Bremer, 1977 and 1987). However, their scopes and formats look like the previous world modeling efforts in their size; variable sets and equations were vast and complicated. In this sense, the tradition of cellular automata computer simulations is distinguished in its status in international relations studies.
Since the tradition of global modeling that focused on the full runs of a world model with many variables, Bremer and Mihalka (1977) developed a theory-driven computer simulation with only a few essential variables. A feature of their simulation is that it is based on the cellular automata framework, in which the geographical configuration among interacting units matters and is to be traced throughout the runs of simulations. These early trials reflected one of the most essential merits of simulation—exploring and tracking the working mechanisms of research targets. With this computer simulation technique, they wanted to investigate the dynamics of world politics modeled by the realist theory. As the basic purpose was not the diagnosis and prescription of world problems but a theoretical, heuristic one, the size of the simulation tends to be compact in the cellular automata simulation. Unfortunately, however, the simulation study was biased toward the realist paradigm. They experimented only with the realist rules, but not with the liberal rules of world politics. This is one of the motivations behind why I would like to use the cellular automata computer simulations in this study.

Bremer and Mihalka’s work (1977) was one of the first efforts to simulate geopolitical world politics on a computer screen—named *Machiavelli in Machina*. Its big puzzle was: “Under what conditions do multistate systems achieve stability, or transform into empires?” Upon this puzzle, the authors conducted cellular-automata-style simulations in order to observe what happens in this two-dimensional space out of the complex interactions among states that follow a particular set of decision rules. The authors tried to investigate the chance of world politics transforming into a balance-of-power situation or into an imperial domination, when each state follows the realist rules. Their study had a nice-looking hexagon-style maps composed of 98 cells in which each

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
cell stands for a state at the initial stage. When more and more wars happen among themselves, these cells merge into multiple-cell states and finally into an empire that holds the whole cells (territories) of the world.

Bremer and Mihalka’s simulated world was designed as an artificial multistate system where the military power of each country and its territorial base matter. Each state knows the geographical positions of its neighbor states and is able to estimate their power levels, with varying degrees of accuracy. The behavioral rules dictate each state to choose a policy among a predetermined set of alternatives in conflict initiation, targeting, coalition, war, division of war spoils, and post-war power adjustment. The system of *Machiavelli in Machina* resulted in a changing pattern of system dynamics over time – from political unification (slow but steady growth of many states by small territorial acquisitions) such as political development or nation building, to balance-of-power, and finally to imperial consolidation. Of course, they found that many runs of simulation ended with the emergence of empires out of the realist mechanism of world politics. As such, Bremer and Mihalka’s model translated the gloomy vision of realism into a visual mode by showing the gradual reduction of the number of states over time. Most of the runs of their simulations made the world collapse into an empire that dominates the world through brutal struggles for power and territories. So their simulation demonstrates the providence – “Let them follow the realist principles, then destiny will be shadowed by the dominance of an empire in many cases!” Thus their simulation shows an important aspect of international relations – the struggle of power – by dramatically diminishing the possibility of any country’s survival rates in a multi-state world system.

41

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Based on Bremer and Mihalka’s pioneering work, Stoll (1987) and Cusack and Stoll (1990) conducted a broader-scale cellular automata simulation to investigate the core mechanisms of the realist world. Stoll (1987) posed two puzzles about the relationship between individual states and international system: “How do states interact to create or modify the system?” and “How does the system constrain the actions of individual states?” Based on the principle of self-interest which is one of the core mechanisms in the realist theory, the runs of his simulation resulted in balancing behavior in some cases and resulted in the emergence of an empire in many other cases. Later simulations by Cusack and Stoll (1990) also investigated the dynamics of world politics by implementing the essential behavioral rules of realism on the computer screen. Their emphasis was on the conflict dimension in which the primary means of gaining power and territory is war.

Cusack and Stoll’s basic version of simulation, called the *Automatic Stabilization Model*, followed the basic scheme of Bremer and Mihalka (1977) by establishing a typical process of war among countries – such as dispute initiation, escalation, war, and power adjustment. In addition, they included another important aspect – civil war – in order to represent the possible disintegration of an imperial state by internal disputes. Also their approach considered the difference between major types of actors – primitive power seeker, power balancer, collective security, and rational actor. Each type of actor follows typical behavioral rules so that it resembles the real-world diversity of decision-making styles across countries. With this scheme, Cusack and Stoll tried to see the changes in the dynamics of system endurance, balancing processes, and the rate of state survival from the war-ridden world politics.

42
Cusack and Stoll's EARTH (Exploring Alternative Realpolitik Theses) model (1994) was a newer version of cellular automata computer simulation to study collective security, which was initiated as an alternate model to the previous realist model. In this simulation, they showed that behavioral rules of collective security among states guarantee higher survival rates for each state than the pure realist rules. In other words, the cooperative behavioral style of states, however small in size, tends to promote the endurance of the whole system multiplicity. This result confirmed Axelrod's (1984) proposition that the strategy based on reciprocity would facilitate international cooperation more than any other strategies. As such, Cusack and Stoll's EARTH model initiated theoretical challenges against the conventional realist paradigm by designing artificial world politics that incorporate the cooperative dimension. However, their research design dealt with the cooperative behavior only in security-related issues rather than in other areas such as trade.

One serious problem in Cusack and Stoll's (1990) simulation, according to Duffy (1992), was its serial assumption in which only one conflict is possible at a time. So, in his alternative model, many potential initiators are selected "concurrently" within each of the simultaneously simulated worlds. For this kind of special simulation, he used a parallel computer called the Connection Machine that has many central processor units and parallel algorithms, while maintaining major features of Cusack and Stoll's framework. His model proposed a "parallel computing environment" which overcomes the problems of the serial assumption. The only difference in research design between his concurrent model and the Cusack and Stoll's previous model was the simultaneity of conflict initiations across countries, which must be assumed for a better verisimilitude of
the real-world conflict processes. This difference led Duffy's simulation to produce quite
different results from Cusack and Stoll's serial model – more impact on the state survival
rate and less impact on the system endurance. As he suggests, parallel computing seems
to be one of the most promising areas in cellular automata computer simulations, but
unfortunately there are many barriers in implementing this method both in hardware and
in software.7

Another contribution by Duffy can be found in his proposal for a strong inductive
method in international relations modeling. His rejection of deduction was based on the
perspective that we cannot trace the “n-actor” problem in a formal manner, so induction
should be taken as an alternate inference method. This epistemology was to be
implemented at the micro-level design that incorporates decision-maker's “satisficing”
tendency (See also Simon, 1985) and local knowledge. These topics are closely
connected to the limitations of the conventional linear framework, which has been the
basis of the realist tradition. Alternate themes in this trend are decision-makers’
misperceptions, nonlinear relationships among war factors, deterministic functions of
power disparities, and localized conflicts. All of these concepts in his research were used
to criticize the “false” assumptions of the traditional deductive approaches – particularly
the realist theory. The raison-d'etre of cellular automata computer simulation can be
found in its role of filling the gap between the conventional deductive method of tracking
the whole paths top-down and the increasing demand for bottom-up way of finding
patterns.

7 The most serious problem is the availability of parallel machines, which kept Duffy's study from being
developed further. Although the speed of CPU has been doubled over and over again for the last years,
parallel machines have not been used much in social science studies. Also the development process for
parallel computing needs special algorithms for multiple processors.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Duffy's CWORLD (1993) is another massively parallel computer simulation of international war. In this model, he hypothesized that history influences the pattern of international competition and cooperation. This model added several elements of history — such as learning (long-term memory as the perception of other’s power capabilities), reciprocity (short-term memory as rewards and punishment over short iterations), and historical friendship and animosities — into his parallel model. Besides, the new model applied the utility maximizing heuristics only when a majority of agents adopt it. By implementing this CWORLD model, he removed the assumption of “ahistoricism” that ignores the real-world historical factors. The result of his simulations showed that the effects of war costs, punitiveness, and the level and variability of misperception on the survival and endurance of states and system, are sensitive to historical factors.

While Bremer and Mihalka (1977), Cusack and Stoll (1990) and Duffy (1992) have stood on the same tradition of international relations simulations for theory tests, Lars-Erik Cederman (1994 and 1997) distinguished himself from this conventional realist’s tradition. The most important theme proclaimed by Cederman was the denial of the traditional assumption of state as an “exogenously given” actor. As an alternative, he presented a model of “inherently history-dependent” actors. Cederman’s approach was based on a wide range of dissident perspectives of critical theory, which the author called “social constructivism.” Social constructivism assumes that social structures, including individual and collective actors and their practices, are all interactively constructed rather than exogenously given. This perspective rejects the conventional rational choice paradigm that emphasizes system equilibrium and methodological individualism (1997,
19-22). Cederman argued that social constructivism is the only relevant paradigm for explaining "emergent" properties of many international relations phenomena.

Cederman's research framework was based on the complex adaptive systems (CAS) theory and heavily used the decentralized computer simulation as a type of "counterfactual thought experiments" (1997, 10). Although CAS simulation sacrifices the deductive power of rational choice model, it has many advantages in treating actors as "dependent" rather than "independent" variables. In this framework, historical junctures fall between the most stable periods; so studying historical discontinuities facilitates the anticipation of future structural transformations of the international system. This position can be understood as a critique of the Waltzian structural neorealism in its questioning of the very origins of the great power system that has been assumed as naturally given by existing theories.

Based upon a micro-level rule-based simulation of world politics, he established the "Emergent Polarity Model (EPM)" in which international state actors are created and macro-level phenomena - such as polarity - emerge from the local interactions of those actors. Cederman's simulation showed that the phenomenon of power politics in international relations, which is a process characterized by low polarity short of universal empire, is a structural outcome regardless of the initial configurations of actors. In addition, he demonstrated that power politics is less likely in defense-dominated system where defensive attitudes tend to increase the chances of state survival, but the likelihood of unipolarity increases significantly at the same time. In summary, Table 2.2 shows the basic research designs and algorithms of each simulation work discussed so far in this section. All of these simulation works used computers and took the cellular automata
framework, even though they are different in their scopes and details. Major research puzzles, simulation configurations, consequences, limitations in research design, and the basic algorithms are compared with each other in the table.

All of the computer simulation works in international relations discussed above conducted experiments on the realist rules in the context of theory-testing, except Cederman who stood for the constructivist paradigm. My study will add more on the existing literature by testing some propositions that are extracted from major theoretical debates, such as war and trade, interdependence-war relationships, and structural factors mentioned in the previous section. There have been many theoretical and empirical works, but no simulations have been conducted to trace the working mechanisms of the theoretical points except for classical realism. On the other hand, existing simulation works have focused only on the realist rules in explaining the structure and process of world politics. Only war (or conflict) was assumed as a state choice, so that those studies have been seriously biased toward the realist thesis. This thesis tries to experiment with the liberal theme building upon the existing realist simulation frameworks. Particularly, I would like to follow the basic simulation format established by Cusack and Stoll (1990) while taking the core algorithms of both the realist and the liberal theories. Thus states in this study will “trade” with each other as well as fight wars between themselves.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Puzzles</strong></td>
<td>Under what conditions</td>
<td>Why do states behave as they do, and what accounts for their survival and success?</td>
<td>What would happen if states’ interactions occur in a concurrent way?</td>
<td>How do states emerge and consolidate their power?</td>
</tr>
<tr>
<td></td>
<td>multistate systems achieve stability, or alternatively, are transformed into empires?</td>
<td>What accounts for the dynamics of the system, and what prevents it from degenerating to the point where it loses its multistate character?</td>
<td>How does decision makers’ reflection (history) impact the survival and endurance of multistate systems and of states themselves?</td>
<td>From where great powers come?</td>
</tr>
<tr>
<td><strong>Landscape</strong></td>
<td>14 x 7 = 98 states CA</td>
<td>16 x 8 = 128 states CA</td>
<td>20 x 20 = 400 states CA</td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>6 neighbors per state (hexagons)</td>
<td>4 neighbors per state (hexagons)</td>
<td>4 neighbors per state</td>
<td></td>
</tr>
<tr>
<td><strong>Actor Types</strong></td>
<td>Measured by military power distributed normally with a mean of 100 and S.D. of 20</td>
<td>Power disparity among states</td>
<td>Power disparity among states</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Knowledge of own and others' power with different level of accuracy</td>
<td>Power management styles - Primitive power seeker - Power balancer - Collective security seeker - Rational actor</td>
<td>Power management styles - Primitive power seeker - Power balancer - Collective security seeker - Rational actor</td>
<td></td>
</tr>
<tr>
<td><strong>Simulation</strong></td>
<td>Empire</td>
<td>Emergence of BOP from self-interested nations</td>
<td>Concurrency leads to more state endurance and less system endurance</td>
<td></td>
</tr>
<tr>
<td>Results</td>
<td>Three phases</td>
<td>Sometimes empires rather than BOP</td>
<td>Emergence of polarity with war and conquest as side effects</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Political unification</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Balance of power</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Empire</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Limitations</strong></td>
<td>Focus only on conflict dimension</td>
<td>Focus only on conflict dimension</td>
<td>Focus only on conflict dimension</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No emergence of new actors</td>
<td>No emergence of new actors</td>
<td>No evolution and learning</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Equilibrium-oriented</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.2: Comparison of Cellular Automata Simulations in International Relations
2.2 Theoretical Propositions

Based on this literature review, I would like to highlight some propositions that are related to the three main puzzles of this thesis. Regarding the first puzzle about the effects of micro-level factors on the system structures, three propositions are established according to existing works on this topic. Particularly, three major dependent variables – system endurance, the balance-of-power, and state survival – will be dealt with using set of explanatory variables. Second, regarding the theme of interdependence and heterogeneity, four propositions will be offered about the impacts of these factors on the probability of war. Lastly, on the topic of structural impacts on the probability of war, propositions will be developed about systemic polarity and the power concentration among major powers, and their impacts on the probability of war.

2.2.1 War and Trade

As reviewed above, Rosecrance's theme (1986) about the territorial world and the trading world is simple, likely and definite. According to him, with more and more interdependence and trade, the more stable world politics becomes. The most probable result of growing economic interdependence might (and should) be a higher level of peace among nations. Another possibility, if any, might be a more plural world where state powers are distributed more evenly. While this theme seems to be very likely and expected, its significance vis-à-vis the conventional realist theme deserves more serious discussion, especially in theory-building and its test through artificial data generated by

49

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
simulations. In the realist tradition, the world has a very high chance of collapsing into a "universal empire" if we model the state only as a primitive power seeker, like Bremer and Mihalka (1977) and Cusack and Stoll (1990). One purpose of my study is to run the simulation of another kind of world with the addition of the rational state with more options – trade as well as war upon the calculation of expected benefits. This is intended in order to see the influence of the change in the unit-level properties, i.e., the introduction of trade, on systemic phenomena. Any comparison of the simulation results between the realist framework – where only security relations are possible – and a mixed one – where both security relation and trade relation are allowed – should deserve our attention in the context of theoretical debates in international relations.

**Proposition 1:** With the introduction of trade, the chance of the system of states to endure rather than collapse into universal empire will be higher than in a world where only security relations (including alliance formation) are possible.

**Proposition 2:** With the introduction of trade, the probability of balance-of-power will be higher than that of the security-only world.

**Proposition 3:** With the introduction of trade, the survival chance of states within systems will be higher than that of the security-only world.
The first proposition deals with the situation in which the number of surviving states varies according to the introduction of the trade option. In the case of Cusack and Stoll (1990), more than 40 percent of simulation runs ended with the emergence of empires. Thus is supported a brutal reality of the realist world on the computer screen. My idea is that we can overcome this pessimistic diagnosis if we model the world with a trade option and that we can identify the conditions that characterize the shape of world politics at the macro-level with simulations. Few simulation studies have been conducted on this point except Bremer and Mihalka (1977) and Cusack and Stoll (1990). Although some real-world empirical data are available back to the early eighteenth century, we may not have enough cases to test these propositions, as discussed before. In this sense, a simulation for those theoretical points might help to confirm the relationship between important variables such as war, trade, and other system-level characteristics. Many worlds can be generated over and over again in simulation studies.

The second proposition focuses on a different aspect – the probability of balance-of-power between states when wars are intended. Previous simulations, especially Cusack and Stoll's (1990 and 1994), have focused on the two modes of balance-of-power – deterrent balance-of-power and defensive balance-of-power. I hypothesize that we may observe more frequent balance-of-power activities (both in deterrent balance-of-power and defensive balance-of-power) in the two modes if we introduce the trading option to state behaviors, which can improve the welfare of both partners. States’ consideration of the potential benefits out of the alternative way of interactions (rather than war) will impact on the states’ own selection of behavior. In the same context, states may be
involved in the balance-of-power activities more frequently than in the security-only world once a war happens for the same reason.

The third proposition deals with the impact of trade on the state survival ratio, which is related to the statistical pattern of an individual states' destiny. Many people may guess that the trade option will extend states' lives in the system. That is, the average number of states surviving up to a certain round of iterations will increase with the trade option allowed in our model of world politics. In this sense, this proposition is not about any individual state's survival ratio but about a systemic one in a probabilistic form.

Altogether, these three propositions become a basic starting point for later propositions, but they are connected to one big topic – the impact of individual states' trade options (between war and trade) on system stability, balance-of-power patterns, and state survival rates. Actually the group of the dependent and explanatory variables in these propositions replicates those of Cusack and Stoll (1990), who implemented simulations only in the security-only world. The reproduction of both their security-only world and the new security-and-trade world will make an interesting comparison regarding the themes of "trading state" suggested by Rosecrance. Chapter 4, as the starting point of simulations, deals with these three propositions.

2.2.2 Interdependence, Heterogeneity and War

Based on the basic propositions suggested above, the next one deals with the impact of interdependence and heterogeneity of dyads on the probability of war. While the unit-level factors are reflected in the basic framework of this study, an extended
version considers the impact of dyadic attributes on the frequency of war. This topic is
related to existing theoretical debates on interdependence and dyadic heterogeneity. The
issue of interdependence was clearly put forward by Rosecrance (1986). His main theme
is that the fundamental factor of the emergence of the trading world is the growing
interdependence among its members. While there are a couple of ways to define and
measure “interdependence,” I will operationalize the concept in an economic sense. 8 The
notion of economic interdependence here implies the existence of an expected utility
between two countries that accrues out of potential trades. If two states have positive
expected benefits out of trade, then they can be regarded as “mutually interdependent” on
the dyad. Actually, this simple concept was used by Cooper (1968) and Rosecrance
(1986). Polachek (1980) also used a similar concept in measuring the level of
interdependence between states.

**Proposition 4:** The more economically interdependent states are, the lower
the probability of war between them becomes.

Regarding this proposition, I will focus only on the probability of war as the
dependent variable in order to match the testable proposition with the mainstream
theories on this topic. That is, I will not use the same set of dependent variables that were

---

8 I would like to briefly mention Waltz’s critique of pure economic conception of interdependence. His
concept of “interdependence as vulnerability” refers to an unequal relationship between countries even
though they are mutually dependent. That is, despite mutually expected benefits from trade, the level of
interdependence may be lower if one country is more powerful than the other (1979, 143-146). Waltz’s
concept, however, is systemic rather than dyadic. He did not suggest any meaningful way of
operationalizing his new concept. So Waltz’s theory of (systemic) interdependence will not be directly used
in this chapter.

53
used in the previous chapter. We can imagine a more peaceful dyad, if the states in the dyad are more interdependent on each other (measured by their expected utility).

However, considering that war also generates an expected utility for a state, we have to test whether and how much economic interdependence leads to a more peaceful world. As it is not possible for us to test this kind of theme in the security-only world, like Cusack and Stoll’s, this proposition enables us to experiment with some challenging topics regarding economic interdependence in this artificial world. A couple of related dimensions regarding economic interdependence will also be considered in testing this proposition in Chapter 5.

Another issue regarding the dyadic relationship – state heterogeneity in terms of size/capabilities – was also initiated by Rosecrance (1986). One of the intriguing themes suggested by Rosecrance is that states tend to rely on trade less as they become larger. This is because, he argued, they become more self-sufficient so that they do not need the requirements for their survival from outside (1986, 14). This statement is closely related to the surge of many small states following the collapse of imperial blocs since World War II. These small countries, according to Rosecrance, are rarely self-reliant, so they “will come to depend on others for economic and even military necessities, trading or sharing responsibilities with other nations” (1986, 15). Rosecrance proposed that this factor is one of the important sources of growing interdependence and dwindling conflicts in the post-War world politics. Therefore, I would like to test the impact of state size on their behavior by asking these questions – “Do big states rely on trade less than small states?” “Do small states fight each other less than big states?” and “Is there any
difference in the conflict patterns between different dyads in terms of state size/capabilities?"

The proposition by Rosecrance may be compared to the seemingly contradictory argument by Waltz (1979) about the trend of interdependence in the post-War era. While Rosecrance put emphasis on the increased number of states and the consequent growth of interdependence among them, Waltz focuses on the decreasing intensity of interdependence among great powers (1979, 145-146). So the impact of size on the war/peace dynamics seems to be inconsistent between Rosecrance and Waltz – the dyad of smaller states is more peaceful for Rosecrance and the dyad of great powers is more peaceful for Waltz. This is an interesting contrast because it is closely related to the theoretical debate on the identification of major factors – the unit-level and the system-level – for the post-World War II peacefulness. If Rosecrance is right, we can interpret the collapse of imperialism and the emergence of many small states as the major factor for the postwar peace. If Waltz is right, polarity – especially bipolarity – can be regarded as the more important factor in the peaceful post-War dynamics.

Waltz argued that the level of interdependence since World War II has decreased. One reason can be found in the attributes of states – they are the same type. This means that the functional similarity of state mechanisms does not encourage the interactions among themselves as many economists argued. A state, according to Waltz, can provide almost everything it needs from inside. The principle of division of labor seems to work more within a country rather than across state borders. Another reason of low

---

9 Of course, we have to remember that the two concepts of interdependence suggested by Rosecrance and by Waltz are different from each other – "sensitive interdependence" for the former and "vulnerability interdependence" for the latter, as discussed before.

55
interdependence, for Waltz, is the nature of structure in world politics: bipolarity reduces the level of interdependence more than multipolarity does. Waltz showed, theoretically and empirically, that states in a bipolar world tend to rely less on others than in a multipolar world (Waltz, 1970, 207-208). Actually, post-war politics have been dominated by two superpowers that have wielded a lot of influence on the ideological as well as material dimensions of their subordinate countries. Waltz focused on the unequal nature of this relationship. Interdependence, according to him, underscores a reciprocal relationship but our bipolar experience has not confirmed this. He argued that the world has been more dependent rather than interdependent among within-camp members since World War II. Across camps, there have been much fewer interdependent interactions. In this context, Waltz tried to reveal the “myth” of interdependence while stressing the unequal nature of international relations (1970, 220-221).

Great powers, Waltz argued, tend to have more resources up to the level of near self-sufficiency. As such, they are relatively less vulnerable to other states (at least within their camps) even though the level of interactions increases. Considering that the concept of interdependence as “vulnerability” presumes opportunity costs when the relationship is broken, we might as well say that superpowers in the Cold War era have been less (vulnerably) interdependent on other countries, while small countries have been more dependent on great powers (including superpowers). The superpowers have controlled small states to a “considerable extent” and this implies that the impact of structure overshadows the beneficial effects of interdependence (1970, 222). As such, Waltz emphasized the importance of state size in international transactions.
Rosecrance talked about the tendency of increasing interdependence among small states, while Waltz emphasized the tendency of decreasing interdependence among great powers. While Rosecrance did not directly mention the pattern of conflict among major countries, his position seems to be in contrast to Waltz's in terms of the probability of war among small states. In Waltz's theory, great powers fight less frequently than small powers; in Rosecrance theory, small powers tend to be more interdependent among themselves than among great powers. As such, I would suggest the following proposition upon the contrast between Waltz and Rosecrance. Here, I would use the term "major" and "minor" for big and small states for empirical reasons. Many studies that used the Correlates of War (COW) dataset used the variable coded "major" for influential states in a given period. Although I will discuss some detailed methodological topics in the research method section of each chapter, I will use these terms to match the empirical portion of my research to existing literature. So "major" powers mentioned here should be understood as referring to the states that hold a certain level of power (three percent in this study) in world politics. No other conditions are assumed beyond the common-sense usage of "major" in COW. Other states that are not classified as major will be identified as "minor" states.

Proposition 5: The probability of war between major states is lower than that between minor states.

This proposition tests both Rosecrance's and Waltz's propositions at the same time. If I find support for this proposition, then we can regard Waltz's argument as more
persuasive than that of Rosecrance in terms of state size and war. In contrast, if I disprove this proposition, then Rosecrance’s liberal theme might be more convincing than that of Waltz’s in the role of state size between interdependence and war. Here another implication of their themes is that the relationship between big states and small states might be different from those of similar-sized states. This case might be more plausible if we cannot accept or deny the above proposition. If so, then there might not be a great difference between the behavioral pattern of big states toward other big states and the one of small states toward other small states. In other words, small states fight each other just as big states fight among themselves. Then, what about the dyad between a major state and a minor state? What about the major-major dyad and the minor-minor dyad? Are the homogeneous states fighting more than the heterogeneous states in terms of their size/capability? In order to test these questions, I suggest the following proposition. This proposition implies that the dyad with a bigger difference in power ratio has a higher tendency of war than the dyad with a smaller difference.

Proposition 6: The probability of war in heterogeneous dyads (in terms of state power) is higher than that in homogeneous dyads.

Although Proposition 5 and 6 are intended to test Rosecrance and Waltz’s seemingly contradictory arguments and a derivative one from those, I have to rely on an arbitrary reference (i.e., COW) for the identification of “major” and “minor” states. This kind of classification makes sense in most cases, but does not show the full information from the data if we use the quantitative dataset. Also the number of dyad types used in
this classification is only four – major-major, major-minor, minor-major, and minor-minor (the first represents the initiator, and the second the target in the dyad). For this reason, I would like to measure the size of states by an ordinal scale rather than by a nominal one. This will allow us to see the impact of size on the state behavior on a continuous scale. Thus I rephrase Proposition 6 with Proposition 7 in order to reflect the ordinal scale in the dimension of state size/capability. That is, we can hypothesize that a higher difference in power ratio between two countries will lead to a higher probability of war between them. Detailed discussions about the relevant operationalization of variables will be provided in the research method section of Chapter 5.

**Proposition 7:** The bigger the difference in power between two states (on a continuous scale), the higher the probability of war between them.

As such, Propositions 4, 5, 6 and 7 test the impacts of dyadic attributes – especially the level of economic interdependence and the heterogeneity in size/capabilities between the states in a dyad – on the probability of war. These tests will be implemented and analyzed in Chapter 5.

### 2.2.3 Polarity, Power Concentration and War

The rest of the propositions are related to the impact of structural factors on the probability of war. The structural neorealist theory – proposed by Kenneth Waltz – has emphasized the importance of *structural* variables in explaining world politics. Of course,
the structural phenomena, such as balance-of-power and polarity, have been the hottest topics even before Waltz. However, Waltz’s structural theory distinguishes itself in its critique of the reductionist approach. Waltz rejected the practice that explains systemic patterns only by lower-level units. In international relations, the state (as a micro-level unit) has been regarded as the most important unit of analysis for systemic patterns; but Waltz tried to overcome the problems of this downward reductionist paradigm. Regarding the war/trade and interdependence themes, Waltz’s theory has some important implications for the main topic of this thesis. If Waltz is right, then we can accept that the phenomena of peace and war are determined more by structural factors, such as polarity, than by micro-level factors. If so, then is there any way to compromise between his structure-oriented approach and the bottom-up one taken by this research?

Waltz criticized the concept of “interdependence as sensitivity” because it ignores many important political aspects. According to him, this is just a simple economic concept based on an unrealistic assumption of perfect competition, so that it overlooks the effects of unequal nature of international politics (1979, 141). He focused on the similarities among state units rather than their functional differences; thus, states as “like units” are not necessarily interdependent with each other. Upon his examination of historical data, unlike the liberals, he drew a conclusion that the level of interdependence has been reduced since World War II. His idea was based on highlighting the difference in capabilities rather than functional dimensions among states. The power distribution, according to him, influences world politics more than economic motives. So he argued that interdependence tends to decrease with smaller number of great powers. Of course,
in his case, the concept of interdependence applies to the whole system level rather than
to the dyad-level (Waltz, 1979, 143-145).

In dealing with Waltz’s theory with regard to my topics, I have to make clear several
points. While Waltz suggested the concept of “interdependence as vulnerability”
implying a relationship that is costly to break, he did not show us how to operationalize it. Therefore, we cannot directly compare Waltz’s political theory and Cooper’s and Rosecrance’s economic theory by the same empirical standard. However, he indirectly mentioned the effects of structural variables (rather than the unit-level variables highlighted by Cooper and Rosecrance), so that we can test his theory with the propositions including these variables, even though I would not participate in the theoretical debate on the meaning of interdependence. What I would like to do here is apply some of his ideas on state power and systemic structures to my simulation scheme, even though his theory covers the whole system in its scope. Here, I would like to measure the impacts of the Waltzian factors on war/peace dynamics with the simulation techniques. That is, I want to see whether and how structural factors impact on state choices of war and trade in this artificial world politics where both options are available.

I also have to tell about the nature and scope of agents (or actors) in his theory. Waltz made clear that his theory is about the structure composed of only major states – or great powers. While he admitted the universality of state functions, he put emphasis on state capability that differentiates one from the others. Particularly, he assumed that world politics is determined by the interactions of major powers. No small countries in world politics hold any theoretical significance in his theory. If a state cannot have impacts on structure, then it is regarded as a passive actor that may be removed from any analytical
scheme (Waltz, 1979, 93-97). This is a great difference between my study and Waltz’s in dealing with the agents of world politics. My theory, based on the complex system theory, regards many small state agents in world politics as necessary elements in generating systemic phenomena, even though they may not directly affect the system at the individual level.

Thus Waltz’s argument, that interdependence will be looser with fewer states, also applies to the relationship between great powers or between the blocs led by great powers. Of course, he praised the virtue of the small number of great powers (or a high level of polarity) in world politics. While the level of interdependence is reduced with the increase of polarity (i.e., with less poles), the world will be more stable with the small number of great powers than with the large number. Therefore, according to his theory, “smaller is more beautiful than small” (1979, 134). For example, a bipolar world will have the lowest level of interdependence – at least in economic sense – between major countries than a multipolar world. Thus Waltz’s theory proposed that a more peaceful world is possible with fewer great powers holding a low level of interdependence between them. This is the point where I would like to test in Waltz’s theory, and it will be compared to Rosecrance’s theory that attributes peace to the growing level of interdependence among many small states.

These points are closely related to the complex adaptive systems (CAS) theory that is the fundamental framework of my simulation. I will discuss major features of the CAS theory in detail in the research method section whenever possible, but it deserves mentioning briefly the difference between the Waltzian structural theory and the CAS paradigm here. At least in terms of epistemology, Waltz squarely contrasts to the CAS
theory. As mentioned before, Waltz rejected the reductionist approach and suggested a structural approach as an alternative. Although he did not use any formal methods, his focus was on the top-down way of analysis for explaining international relations. On the other hand, the CAS theory stands for the opposite direction in that it prefers to analyze systemic phenomena “bottom-up” while emphasizing the interactions among units in the structure rather than the units themselves. While the CAS approach itself denies the reductionist approach too, it does not encourage the “top-down” approach. All systemic phenomena are regarded as “evolving” from the interactions of units rather than “given,” unlike Waltz’s assumption. Therefore, structural variables in the CAS approach are mostly regarded as dependent variables, despite the feedback effects from the structure.

I do not want to get involved in any epistemological debate here, but it seems to me that the comparison of two perspectives will help us clarify a major difference in each theoretical argument. I want to test whether the structural factors impact on state choices about war and trade in my artificial world (the security-and-trade world where both of the realist elements and the liberal elements exist). Therefore, I would like to see how much impact structural variables, as Waltz argued, have on individual actors’ choice and its macro-level patterns (of war). If we can find significant impacts, then we have to consider revising our model of world politics with more structural variables as Waltz recommended. Or, we may have to compromise the contradictory epistemologies of “top-down” and “bottom-up” approaches. Perhaps, we may need to consider any feedback effects from the systemic environments back to the unit level. Whatever direction it implies regarding this test, it seems to me that structural variables deserve to be included in our test at least for a heuristic purpose.

63
As such, Proposition 8 tests the most important structural factor – polarity – on the probability of war. Is there any difference between bipolarity, multipolarity, and unipolarity in the frequency of war? If so, how much difference is there between those different types of polarity? If polarity influences state behavior, is it the same or different across various types of dyad? Do major powers fight each other more under bipolarity than under multipolarity? Particularly, I would like to see whether polarity influences states’ choices differently between major countries and between minor countries. Thus Proposition 8 mentions a hypothetical negative relationship between the polarity of world politics and the probability of war. The level of polarity is measured by a specific algorithm (this will be discussed in detail in Chapter 6), but roughly it rises as the number of major powers goes down with more inequality among themselves.

Proposition 8: The probability of war declines as the number of poles decreases.

While the factor of polarity seems to be a good indicator for structural attributes, it may have problems in empirical contexts as it applies the same weights on different poles. Edward Mansfield (1994) offered this point by arguing that poles are not equivalent and that non-pole major powers also have influence (Mansfield, 1994, 72-73). I agree with his critique, so that this thesis tries to use his index of power concentration (devised by Ray and Singer) as another explanatory variable for the same relations between the explanatory variables and the dependent variable, in addition to the ordinal concept of polarity. Mansfield’s elaboration of the power concentration index reflects the
missing element of inequality among major powers and the role of major powers that are not poles. Using this index, he argued that the relationship between power concentration and war frequency follows an inverted U-shaped pattern, unlike monotonic arguments found in existing theories (1994, 80-81). Considering this point, I would like to test the same theme with the following proposition:

**Proposition 9: The probability of war follows an inverted U-shaped curve as the level of power concentration increases among major states.**

Mansfield’s usage of power concentration will be discussed in Chapter 6, but it is worthy of mentioning here about its basic feature. The level of power concentration is a continuous variable, unlike polarity, so that it shows a different aspect of the structure of world politics by focusing on the inequality among major powers. Mansfield’s theory is based on the assumption that the level of concentration among major powers is related to war frequencies. The result of his analysis shows an inverted U-shaped pattern so that we expect a lower frequency of war at lower and higher concentration levels, while there is a higher chance of war at the intermediate level of power concentration. As his theory is closely related to the systemic pattern of world politics but with a different logic and measurement, I would like to test his theme, too. It will be a good contrast to the Waltzian theory of polarity and war. As such, these themes on the effect of structural factors – polarity and power concentration – on the probability of war will be tested in Chapter 6. The overall relationship between the main puzzles of this thesis and the propositions based on existing literature is summarized in Figure 2.1.
Proposition 5: The probability of war between major states is lower than that between minor states.

Proposition 8: The probability of war declines as the number of poles decreases.

Puzzle 3 (Chapter 6)
How do the systemic properties affect the probability of war in world politics?

Proposition 6: The probability of war in heterogeneous dyads (in terms of state power) is higher than that in homogeneous dyads.

Proposition 7: The bigger the difference in power between two states (on a continuous scale), the higher the probability of war between them.

Proposition 9: The probability of war follows an inverted U-shaped curve as the level of power concentration increases among major states.

Figure 2.1: Organization of Propositions
CHAPTER 3

RESEARCH METHODS

This study is based on computer simulation, which has become a popular research method in political science in addition to formal and statistical methods. In particular, this study uses a combined format of simulations from two existing literature – one from the conflict model of Cusack and Stoll (1990) and the other from the trade portion of Epstein and Axtell’s Sugarscape model (1996). A brief introduction of the combining process of these two main modules will be discussed in this chapter, but the detailed setting of variables and equations will be dealt with in the research design section of Chapter 4, 5, and 6, according to their connection to each theme. In addition, in this chapter, a short discussion on the tradition of cellular automata simulation, object-oriented programming and agent-based modeling will be done before the survey of main algorithms.

3.1 Cellular Automata Simulations

The tradition of cellular automata simulations was not rooted deep in mainstream international relations studies before Bremer and Mihalka’s pioneering work (1977). The failure of large-scale simulations in the previous period in international relations as well
as economics seems to have great impacts on the later simulation projects. Also, before the introduction of powerful desktop computers, any computer simulation was not an easy job to implement because of expensive running costs. In some portions of academy, simulation has been regarded as a "hyperreal" phenomenon that is a side effect in the age of technology (Der Derian, 1988). Nevertheless, the tradition of cellular automata simulations has remained alive since the introduction of Bremer and Mihalka.

Furthermore, with the development of hardware and software, the versatility and the efficiency of computer simulations have dramatically increased these days. Let's discuss why we need simulations, especially the cellular automata type, for international relations studies.

Simulation methods, particularly the computer-based ones, stand between the two mainstream methods – induction and deduction – by combining essential elements of both. According to Robert Axelrod, simulation is a "third way of doing science" beyond the contradictions between induction and deduction. Simulation studies do not try to prove theorems but generates data to be analyzed. However, its data generation is based on a "rigorously specified set of rules" rather than directly imitating or measuring the real world (Axelrod, 1997, 3). We do not need to build a deductive, mathematical model to every detail of the target we are interested in, but we can directly write a program that manipulates data in a real-world way. Also simulation lets us handle the qualitative aspects of research targets, but upon the consistent application of logical causations. One of the best merits of the simulation method is the generation of individual-level data, which is not available in many empirical data that are aggregated and averaged (Clarkson and Simon, 1960, 925). In this sense, simulation studies can be a good alternative...
whenever we need to compromise between the small-N problem and the strict formal logic which is too abstract for the real-world matters.

Cellular automata simulations have been used as a dynamic model since Stanislaw Ulam’s original idea that the computer can print out ever-changing patterns given certain fixed rules (Poundstone, 1985, 14-15). The basic idea of cellular automata is that any recursive application of simple rules with a computer may generate complicated patterns that defy mathematical analysis. It models “discrete dynamic systems of interacting units, where the units are very simple and the rules of interaction local” (Taber and Timpone, 1996, 25). Units (also called cells) in cellular automata can be represented by very simple behavioral rules, but the consequent systemic phenomena go beyond any mathematical analysis, such as fluid turbulence. The mechanism of this type of system can be described by the simple construction of cellular automata in which components generate the behavior of “essentially arbitrary complexity” (Wolfram, 1984, 194). This advantage of cellular automata – representing complicated systemic phenomena with simple rules – makes it a promising tool for analyzing complex research targets even in the social sciences, such as social matters, economies, and world politics where the observed phenomena are featured by complexity but theoretical rules and mechanisms need not be.

Cellular automata simulations have been popular since John Conway’s “Game of Life” in 1970 which showed that very complicated, unpredictable patterns emerge from the implementation of very simple rules (Poundstone, 1985, 24-25).\(^\text{10}\) A cell in his game

\(^\text{10}\) Conway’s idea on cellular automata was introduced by Martin Gardner in his *Scientific American* column and took attention of the public (Gardner, 1970, 120-123).
has only one of two states—on or off—according to a couple of simple rules. On a large
lattice of space, any initial combination of these cells that apply local rules at the same
time generate diverse patterns of on and off and even dynamic, moving patterns, too. The
"Game of Life" shows that the only way of discovering the emerging patterns in this kind
of complex systems is the computer simulation.\(^\text{11}\) The "life" in the "Game of Life" is an
example of a cellular automaton in that it is alive (or not) and thus dynamic (Sigmund,
1993, 13). So, without any explicit programming of the conditions for each cell for the
specified time span, the whole group of cells in the system changes their statuses
automatically. At the systemic level, however, we can observe a pattern which is not
programmed at all. Furthermore, the agent's rules are local, meaning that each cell (or
agent) follows the rules without knowing any global information and without considering
the effects of its own behavior at the systemic level.

Thomas Schelling's segregation model (1978, Chapter 4) was another cellular
automata simulation that modeled the bottom-up process of social phenomena. He
showed us, with his model, that ethnic segregation can emerge out of individuals' simple
behaviors even though nobody intended the consequent phenomenon. In his model, an
individual (or a cell) has eight neighbors around her. She is content or discontent with the
configuration of neighbors according to the basic level of demands, such as colors of the
neighbor occupants. If she is discontent with the color of neighborhood, then she will
move to the nearest empty space for her satisfaction. If we start with a random

\(^{11}\) Conway's rules are as follows: For a given cell, there are eight neighbor cells (like a checkerboard).
When the number of neighbors which are "on" is exactly two among eight, the cell keeps on its current
status into the next generation. If it is exactly three, the cell will be "on" in the next generation, regardless
of the cell's current status. In other cases (when the number of "on" neighbors is 0, 1, 4, 5, 6, 7, or 8), the
cell will be "off" in the next generation. There are no other rules in the game (Poundstone, 1985, 26).
distribution of various colored-groups and apply the rules to each actor, then every actor will select a new environment and it will influence the environmental patterns through a chain reaction. The consequent pattern of segregation emerging from this chain reaction can be interpreted as the “observable aggregate phenomena” that could be compatible with “molecular movement” (Ibid., 150-153). As such, Schelling’s simulation model for segregation in society showed us that unexpected macro-level pattern can emerge from the simple rules of the micro-level agents.

As reviewed in the beginning section of this chapter, Bremer and Mihalka’s simulation (1977) was the first simulation effort to apply this cellular automata framework to world politics. Later works by Cusack and Stoll (1990), Duffy (1992), and Cederman (1997) also took cellular automata as the basic framework of their computer simulations. The cellular automata framework has a spatial lattice of cells, represented by individual units and their neighbors (of the same type). Each cell has a finite number of states (on or off, or any other discrete type of conditions), which can be arranged by researchers according to their interests. The state of each cell is updated simultaneously upon the process of discrete time schedule. In each time period, cells behave according to local rules given by programs. The locality of rules in cellular automata is the most important feature that makes this tool fit into the CAS theory. That is, all cells (as actors) behave but their actions apply or affect only their neighbors. Nevertheless, it generates “patterns” at the systemic level even though these systemic patterns are not programmed in advance. In this way, cellular automata simulations embody the principle of “bottom-up” mechanisms we are interested in and many social phenomena have been modeled by this type of simulations.
The success of cellular automata simulations has been due to its emphasis on the "mechanisms" rather than on the "parameters" of the target as our main interest. In other words, the fundamental purpose of cellular automata modeling is not the "prediction" of real-world phenomena throughout the calculation of feasible scope of parameters, but the "heuristic investigation" of essential features of our research targets. This distinguishes the cellular automata simulations from econometric simulations which are interested more in the estimation of specific parameter values. The old tradition of global modeling in international relations (such as the Forrester-Meadows Model and the Bariloche Model) was intended to solve the "world problematique," which was suggested by the Club of Rome, by covering most fields of human behaviors such as environment, development, information, technology, education, etc. Nevertheless, its result was pessimistic and had to face the critique for excluding the political factors and for its "too broad" scope. As Doran and Gilbert put it, the project of global modeling fell into the "trap of verisimilitude" that puts "plausible detail into the model program not because it is required, but because it is plausible" (1994, 13). The effort for GLOBUS was an alternative for the problems in the previous simulations by including politics and state-oriented framework (Bremer, 1987). Cellular automata simulations do not intend to achieve this big, ambitious goal; instead it narrows down its aim into a heuristic role.

Bremer, in his earlier study of simulations, put this point clearly. According to him, simulation should be a "structure-driven" (i.e., mechanism-driven) model that focuses on the estimation of parameters based upon the more fundamental modeling of

---

12 This is quite similar to the "trap of tractability" in mathematics, which is to "subordinate everything to being able to do something analytic with the model," according to Doran and Gilbert (1994, 13).
structure. This is sharply contrasted to the "parameter-driven" econometric simulations (Bremer, 1977, 94-95). In the structure-driven method, the selection of "influential" factors is an essential procedure, while the parameter-driven method tends to comprehend all available factors so that they become immediately overwhelmed by the huge dataset. In this way, the cellular automata simulations intend to explore the internal working mechanisms of the target system by comprehending only effective factors, which was not intended in the old simulation studies. In particular, in the system of complex interactions among members, the macro-level phenomena cannot be explained by conventional differential equations or any other mathematical tools. The method of cellular automata simulation is an excellent alternative in this case.

Figure 3.1: Cellular Automata Structure of World Politics

---

Bremer argues that "structure" can be understood as the "form of the relationships between variables, and this must be assumed in order to estimate the unknown parameters" in international relations studies. Bremer cites Jay W. Forrester in distinguishing "parameter" and "structure" for discussing simulation methods. As Bremer mentions, a parameter is easy to change without reprogramming or recompiling in a technical sense; on the other hand, the structure of the model is hard to change or manipulate in the actual programming. (Bremer, 1977, 96).
The illustration in Figure 3.1 shows the cell structure of the original cellular automata simulations used by Bremer and Mihalka (1977). Cusack and Stoll (1990) also took the same format, while later cellular automata simulations used different formats such as the lattice-type board used by Duffy (1992) and Cederman (1997). Unlike the conventional cellular automata simulations, the use of honeycomb-style structure with six neighbors has some advantages in the representation of neighboring pattern; it is more complex and closer to reality than the conventional, simple four-neighbor structure. Of course, the cells are organized both by columns and by rows. The total number of cells is 98 (14 columns by 7 rows), so that it is quite enough to embody the large number of agents in the system.\textsuperscript{14} This thesis will use this six-neighbor format again, while modifying the behavioral rules and the attributes of individual agents.

3.2 Object-Oriented Programming and Agent-Based Modeling

One important trend in simulations these days is the popularity of object-oriented programming (OOP) techniques. OOP is based on the perspective that our world is a complex system and that we humans have limitations in dealing with this complexity. One solution for this limitation is to narrow down the scope of our interest from the whole system (which is our interest) to the “units” of the system which include functional relations among themselves. OOP thus can be defined as a “method of implementation in which programs are organized as cooperative collections of objects, each of which

\textsuperscript{14} This point must be so in the period when Bremer and Mihalka invented this scheme; the total number of state actors at that time was slightly over 100.
represents an instance of some class, and whose classes are all members of a hierarchy of classes united via inheritance relationships” (Booch, 1994).

The basic philosophy of OOP is to devise the simplest objects of the real-world object that has attributes (variables) and related behaviors (functions) that are relevant to research problems, and then let those objects interact with each other (Fishwick, 1995). This method has been proven stronger than the conventional procedural programming in simulating complex systems that does not allow any formal analysis. All we have to do in OOP is program objects, make them interact, and then measure the consequences of those interactions. Therefore, the best applications for OOP are “those that are likely to require ongoing change and are inherently complex” (Entsminger, 1990, 99).

In a technical sense, OOP is characterized by many new concepts – such as data abstraction, encapsulation of information, polymorphism, modularity, and hierarchy of abstraction. All of these new techniques have been introduced to improve the technical efficiency of programming. These techniques also help in organizing the complex relationships between components of programs in a hierarchical, integrated, and organized way. However, the most appealing case for OOP in contrast to the conventional procedural programming is that it resembles the target better if the system is complex. For example, the procedural programming technique should always check the parameters at the global level in order to get the values we want. Thus, the values of “variables” in procedural programming tend to be separated from “functions” that change

15 Technically, the object in programming means the implementations of class which is a user-defined type. For example, when we simulate a world of states we define a class type called state. Once we define a class, we generate as many objects as we want in the running of simulations. That is, individual cases of state type are generated from the molding of class. “State” is regarded as a “class,” while “USA,” “Japan,” and other individual cases of state are regarded as “objects.”
themselves. On the other hand, OOP combines variables and functions together in the definition of an object, so that it need not check the changing values of interested variables during the run of simulations. OOP defines and generates objects (appropriate to each research topic) that hold variables and related functions, letting them interact with each other, and then collecting parameter values from those objects after simulation. The research target is closely connected to associate functions in OOP, so that the overall running of simulations looks more like the real world. Among many OOP languages, I will use C++ which has recently become popular for general-purpose programming. Unfortunately, previous cellular automata simulations were written with procedural languages, like Fortran or C, so that a new writing process was required for me to implement the OOP philosophy.

Regarding the OOP technique, it is worth mentioning that agent-based modeling (ABM) is a specific form of the OOP. ABM is based on the CAS framework in which agents are interconnected in a network, and a dynamic, aggregate behavior emerges from individual activities (Holland and Miller, 1991). Genetic algorithms and classifier systems can be regarded as types of ABM. No centralized design is assumed in this framework, so the focus has moved from the level of structure (in conventional modeling) to the level of agents, especially to their interactions. It is, in this sense, ideal

---

16 Suppose we are programming the simulation of state interactions with war and trade functions. We are interested in the values of state welfare after these behaviors (functions). In the procedural programming, the value of each state’s welfare level is represented by a variable. Groups of functions, such as war() and trade() are written separately from those variables, even though they change the parameters of those variables. Therefore, we have to define all the behavioral functions, whether they are applicable to each variable or not, at the global level. On the other hand, OOP combines state and related functions – war() and trade() – into the definition of state. Once we define a type of data called state, it holds function definitions in itself at the same time. Then we generate a number of state objects, make them interact with each other, and the result of interactions will be saved into the variables of each state object or into a parameter variable.
for modeling a decentralized and massively parallel “microworld” (Resnick, 1994). In this ABM, like the real world system, the parameters of the system vary during the simulation rather than being fixed in advance. It is argued that ABM has more advantages than the traditional simulation method in implementing “emergent” social structures and group behaviors from the interaction of individuals that follow specific rules under bounded rationality.

The philosophy of ABM is closely related to the CAS paradigm and one of its major tools – OOP (Holland and Miller, 1991; Epstein, 1999). In addition, there have been a lot of developments related to this framework, especially in the programming technique called agent-oriented programming (Shoham, 1993; Boutilier, Shoham and Wellman, 1997). While this thesis does not intend to survey all of these new trends, it should be enough to mention that these new techniques share the philosophy of the “bottom-up” framework with the CAS theory, in which systemic, global patterns are generated by individuals’ interactions rather than assumed given or predetermined in advance. As such, I would like to reflect these two traditions of OOP and ABM in terms of programming techniques and the philosophy of bottom-up epistemology in programming and running my simulations.

---

17 According to Epstein, the agent-based model is especially powerful in “representing spatially distributed systems of heterogeneous autonomous actors with bounded information and computing capacity who interact locally.” In his perspective, the canonical ABM follows the principle: “Situate an initial population of autonomous heterogeneous agents in a relevant spatial environment; allow them to interact according to simple local rules, and thereby generate – or ‘grow’ – the macroscopic regularity from the bottom-up” (1999, 42).

---

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
3.3 Algorithms: War and Trade

3.3.1 Basic Module

The basic algorithm for this study starts from the replication of Cusack and Stoll's simulation design (1990) which is one of the most comprehensive cellular automata frameworks ever developed in international relations. The basic flow of Cusack and Stoll's model begins with the parameter setting before the main modules. The parameters in their simulation design are divided into two groups - system parameters and state-specific parameters. Systemic parameters include the standard deviation of power distribution, the sigma value for the likelihood of victory, war cost maximum, disproportionate war cost ratio, and reparations rate. State-specific parameters include power management styles, decision-making routines, power assessment error ratios, power growth ratios, and maintenance costs. The assignment of parameters in this study will use these two groups, even though there are some modifications. For example, civil war related parameters and decision-making styles will not be considered in this study because these modules are not included. Also the parameters that are not appropriate in case of the rational actor will be removed or fixed to constants, such as the sigma for the likelihood of victory and the war-tied parameter. Details of the configuration of included/excluded variables between Cusack and Stoll's original simulation and this study will be discussed in Chapter 4.

The actual run of Cusack and Stoll's simulations starts with the civil war module in each country. For several reasons, as discussed before, the civil war module and the
The decision-making types will not be considered in this study.18 The inter-state conflict module follows, by choosing an “initiator” of conflict upon the relative power share of each country in the system. Cusack and Stoll’s simulation follows one of two rules according to the decision-making styles. The first is a general rule for most decision-makers – Choose a state randomly, but with the weight of each state’s relative effective power. Effective power is calculated by real power subtracted by maintenance cost. The second rule only for the rational actor needs a more complex calculation. First of all, each state calculates her expected utility of war with each neighbor state. Then a state will be chosen randomly with the weight of relative expected utility among the pool of states that have positive expected utility. In this case, any state having no positive expected utility will not be considered as an initiator. The selection process for rational actors will be determined by the following equations:

\[
EU_i = p_{nw} \cdot PGAIN_{nw} + p_{nl} \cdot PLOSS_{nl} - WARCOST_i \cdot EPOW_i
\]

\[
p_{INIT_i} = \frac{\max(+EU_i)}{\sum_{i=1}^{n} \max(+EU_i)}
\]

where \(p_{nw}\) and \(p_{nl}\), respectively, are expected probabilities of win and lose; \(PGAIN_{nw}\) is the power gain that would accrue to state \(i\) if she wins the war; \(PLOSS_{nw}\) is the power loss she

---

18 The most important reason for removing the civil war module is the size of simulation. Cusack and Stoll’s original simulation was very big, so any replication of theirs with new programming is a dramatically complicated task. So I decided to extract only the essential core from their project – the conflict/war module including alliance formation – while removing some modules. As the main focus of my study is the comparison of the security-only and the security-and-trade worlds, domestic conflict module and decision-making style were the first candidates to be eliminated from this study.
suffers when she loses the war; $WARCOST_i$ is the ratio of war cost for state $i$ whenever a war happens; $EPOW_i$ is the estimated power level of state $i$; and $p_{INIT_i}$ is the probability of state $i$ to be selected as the initiator (Cusack and Stoll, 1990, 78-79).

In this study, the selection of the initiator will follow the same algorithm for rational actors like the above one suggested by Cusack and Stoll. The only difference is that the state agent in my study is a "rational utility maximizer" both for war and trade, so that she will compare the expected utility of war and trade at the same time for all of her neighbors. Then, she will be inserted to the candidates' pool for selection if she has any positive expected utility, regardless of its choice (war or trade) and the size of expected utility. The selection process is also determined by the uniform random number generation weighted by each state's relative capability. Here, I would use the relative power of each country as the weight rather than the size of expected utility. Thus a state with positive expected utility (for war or trade toward one target of her neighbors) has higher probability of being selected as the initiator at a time if her relative power share is larger in the system. Actually, every state must have positive expected utility at least for trading behavior because trade is assumed to generate benefits for both sides. However, due to the effects of size difference on a trade dyad, I set a limit for the minimum size of trade unit.\footnote{Trade unit is determined by the following algorithm: first, find the minimum between the expected utility of two trading partners; second, if the minimum is more than 1.0, then the size of trade unit between the two countries will be set to 1; third, if the minimum is less than 0.00001, then the size of trade unit is set to zero; finally, discrete values of 0.0001, 0.001, 0.01, and 0.1 are selected as the basic trade unit of resources between partners according to the size of minimum expected utility between the two. A big state, when considering to trade with a very small neighbor, may have zero expected utility of trade.}

This will generate many situations in which a big state does not have any positive expected utility toward a very small trade partner.
If an initiator is selected, then her action is selected automatically according to her expected utility. In case of trade, I assume a simple exchange or resources, of which the process is described in the section for the trade module. If the initiator chooses a war against one of her neighbors, then the target decides whether to fight or to make bids for a defensive alliance. If she thinks that the probability of her victory is bigger, then she does nothing and the war module begins. Otherwise, she makes bids to the neighbor countries of the initiator for a defensive alliance. The candidates for the defensive alliance accept or deny the target state's bids according to the same logic as that of the initiator. That is, a state will join the defensive alliance if the alliance has positive expected utility; otherwise, she will deny the bid. If no defensive alliance is formed, then the conflict starts as the initiator planned.

If any defensive alliance is established, the initiator compares her power with that of the defensive alliance. If she finds that her power is more than that of the defensive alliance, then she goes to war. Otherwise, she has a chance to form an offensive alliance by making bids to the neighbor states of the target. If no offensive alliance is formed, then the initiator gives up her intention of war and the simulation iteration stops and the next one begins. Otherwise, the process power comparison and calculation of winning probability goes on between the defensive alliance and the offensive alliance. If the target state calculates that the probability of the defensive alliance's winning is higher, then the war module starts. Otherwise, she has another chance to expand the defensive alliance by making bids to the neighbor states around the initiator, who have not yet joined any alliance. After this process, the war process begins. Once a war happens between states, then its consequence depends on the real power difference between the two sides and a
random factor. After the war, the amount of territorial transfers and indemnities from the losers are calculated and distributed to the winners. Also the power level of each state is recalculated upon the transfer of territories and indemnities. Then the iteration ends and a new one begins. The algorithm of Cusack and Stoll’s basic module is summarized in Figure 3.2. Also Figure 3.3 and Figure 3.4 show the details of the initiator selection process and the conflict escalation process (through alliance formation), respectively.
Figure 3.2: Cusack and Stoll's Basic Algorithm
Start the selection process (Pool of states)

Each state calculates $EU_{\text{trade}}$ and $EU_{\text{war}}$ for all contiguous neighbors

Insert the state with positive $EU$ (war or trade) into the pool

Size of pool $> 0$?

End of iteration (Go to the next iteration)

YES

Generate a uniform random number

Sort the pool from the lowest to the highest

Select the initiator from the pool with the random number weighted by power

Initiator's max. $EU$ is of war?

NO

The trade module starts

YES

The war module starts

Figure 3.3: Initiator Selection Process
Start the conflict process (initiator and target selected)

Candidates of defensive alliance > 0?

YES → Target calculates EU of each combination of possible defensive alliance

Any ally possible with +EU for target?

YES → Target makes bids to the possible defensive alliance with max. EU

Members of the potential alliance calculate EU of alliance to join

Any candidate accepts the bid?

YES → Initiator calculates the power of the defensive alliance

Estimated defensive power > initiator’s?

YES

NO

NO

YES → Target makes bids to the possible enhanced alliance with max. EU

Members of the potential alliance calculate EU of alliance to join

Any candidate accepts the bid?

YES → Target makes bids to the possible enhanced alliance with max. EU

Members of the potential alliance calculate EU of alliance to join

Any candidate accepts the bid?

YES → Target makes bids to the possible enhanced alliance with max. EU

Members of the potential alliance accept or deny the bid to join

Any ally possible with +EU for target?

YES → Target makes bids to the possible enhanced alliance with max. EU

Members of the potential alliance calculate EU of alliance to join

Any candidate accepts the bid?

YES → Target calculates the power of the offensive alliance

Estimated offensive power > target’s?

YES

NO

NO → War starts

Initiator gives up war

Candidates of defensive alliance > 0?

NO

Candidates of offensive alliance > 0?

NO

Candidates of enhanced alliance > 0?

NO

Initiator calculates EU of each combination of possible offensive alliance

Any ally possible with +EU for initiator?

YES → Initiator makes bids to the possible offensive alliance with max. EU

Members of the potential alliance calculate EU of alliance to join

Any candidate accepts the bid?

YES → Initiator calculates the power of the defensive alliance

Estimated defensive power > initiator’s?

YES

NO

NO

YES → Initiator gives up war

Figure 3.4: Conflict Escalation Process
Cusack and Stoll experimented with four distinct types of decision-makers, but their basic run was composed only of primitive power seekers, who choose attack whenever it has a better chance of winning, regardless of war costs. Types of decision-makers were experimented in an extended version of their simulation (1990, Chapter 5), but I would replicate their simulation only with the rational utility maximizer type. Only the rational utility maximizer can make a consistent decision between war and trade, so that other types of decision-makers will not be considered in the combined model of “security-and-trade world” in this study. Also, as mentioned before, I decided to eliminate the civil war module and some other mechanisms from Cusack and Stoll’s Realpolitik model. Other modules remain in my replication.

Upon this basic structure of simulation processes, let’s briefly discuss on the parameter settings. As I mentioned above, I would not use system-level parameters as far as I can, in order to keep the bottom-up style framework. However, whenever necessary, I did use some in order to compare my security-and-trade world with the conventional security-only world. Five major parameters – power distribution, power estimation error, war cost maximum, reparations, and disproportionate war cost ratio – will be assigned at the initial stage of simulation runs, while fixing the likelihood of victory (LV) sigma\(^{20}\) value to 1.0 (like Cusack and Stoll’s rational actor module). I would like to use these major variables in my simulation and compare the impacts of them to the systemic

\(^{20}\) The likelihood of victory (LV) sigma is an important parameter that determines the victory in a war, upon the relative power of two opposing sides. Cusack and Stoll set the default value of this to 1.0, 3.0 and 5.0. When this value is 1.0, the winner is determined mostly by the relative power between countries. If this is set to 5.0, then the winner of the war is determined more by the chance factor than the relative power (See Cusack and Stoll, 1990, 76-77). The impact of the LV sigma was the highest among all the explanatory variables in my pre-runs of simulation. The reason to fix this value to 1.0 will be discussed later.
phenomena between the security-only world and the security-and-trade world. Also these
major variables will be used in examining the relationship between economic
interdependence and war, and between structural factors and war in Chapters 5 and 6.

In addition to these major system-level variables, I will set another parameter of
the standard deviation in metabolism distribution, which works like the standard
deviation in power distribution. This variable is introduced here in order to make the
trade demand level in a state vary within ranges, so that I can measure the impact of its
variation on the system-level phenomena. The range for the standard deviation of
metabolism distribution is set to between 1.0 and 6.0, like that of power distribution. The
parameter of resource growth rate is also randomly assigned within a relevant range
(0.0 – 0.005) as of Cusack and Stoll’s power growth rate. This parameter will not be used
in the empirical analysis, while it is inserted in the simulation implementation. Also, as in
Cusack and Stoll’s original scheme, the calculation of one’s own power or other states’
power is prone to error, based upon the parameter of the standard deviation in error
distribution. Each state’s error rate in power calculation follows a normal distribution
with a mean of 0.0 and the standard deviation which is randomly assigned (from a
uniform distribution) at the initial stage of runs between 0.1 and 0.3.

At the initial stage of simulation runs, each of 98 cells is assigned a certain level
of resources. While Cusack and Stoll used the notion of “power” for these territory-based
units, I decided to use “resources” in order to combine the trade module with their
original framework. That is, a cell is assigned a certain level of “resource A” and
“resource B” of which values range between 0.0 and 1.0 in terms of its share of the total
system resource A and resource B. As each state (whether one-cell state or multi-cell

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
state) is also assigned metabolism rates for resources at the initial stage of simulation runs for resources A and B – the rate of internal demand for each resource – a state (composed of one cell or multiple cells) has a certain level of satisfaction – called “welfare” in economics. I will use this concept as the index of state power. Imagine the amount of resources, \( w_1 \) and \( w_2 \), and metabolism ratios, \( m_1 \) and \( m_2 \), and \( m_\text{ras} \) as the sum of \( m_1 \) and \( m_2 \), of a state. Then, the welfare function (or the power function) of a state that produces a relative level in a system is (Epstein and Axtell, 1996, 97):

\[
W(w_1, w_2) = w_1^{m_1} w_2^{m_2}
\]

This equation seems more complex than the old concept of power, but it enables us to combine the trade module with the conventional realist model because it generates a consistent index of power that can be used for war as well as for trade. Here, state power is determined by the amount of resources, internal demands (metabolisms) for those resources, and the consequent satisfaction level from the combination of these factors represented by the value of \( W \) in the equation. Before the selection of the dyad of initiator-target, and after the war/trade module, each state in the system recalculates her power level by this equation.
3.3.2 War Module

Once a war starts, it should end with clear identification of winner (or the winning group) and loser (or the defeated group). The determination of winner is based on the following equation (Cusack and Stoll, 1990, 76):

\[ LV(t) = \frac{1}{\sqrt{\pi\sigma^2}} \int_{\infty}^{\infty} e^{-\frac{z^2}{\sigma^2}} \, dz \]

where \( LV(t) \) is the likelihood of victory for a state whose power ratio relative to the other state is \( t \), and \( \sigma \) is the control parameter for the likelihood of victory that varies between 1.0 and 5.0 (See details in the research design section of Chapter 4). As discussed before, the value of \( \sigma \) is fixed to 1.0. This makes the war determined more by real power differences between two fighting units, rather than by a chance factor.

Figure 3.5 shows the relationship between relative power, the likelihood of victory and \( LV \) sigma. As the power ratio of this side \( (i) \) to the other side \( (t) \) gets bigger, the chance of winning by this side increases. When the power ratio stays around 1.0, then the likelihood of winning by this side is about 0.5. However, a slight advantage in power ratio guarantees the winning with the small value \( (i.e., \ 1.0) \) of \( LV \) sigma. This function is also used in the calculation of expected utility of war when a state decides its behavior with a neighbor state, at the initial stage of each iteration. It is also used in the calculation of expected utility of alliance.
After a war is over, every participant should pay for the war cost, regardless of war result. The cost of war for an individual state is calculated by the following equation (Cusack and Stoll, 1990, 87):

\[
WARCOST_i = \left(1.0 - \frac{(LSR - 0.5)}{0.5}\right) \times WARCOST_{\text{MAX}}
\]

where \(LSR\) is the ratio of the larger side’s power to the smaller side’s power and \(WARCOST_{\text{MAX}}\) is the system-level parameter that is randomly assigned at the start of each simulation run. Here, we use a weighting factor for the disproportionate war cost based on...
on relative capability ratio. Thus, the real payments for the cost of war for the stronger weaker sides are calculated by the following equations, respectively (Cusack and Stoll, 1990, 88):

\[
\begin{align*}
\text{WARCOST}_{\text{stronger}} &= \text{WARCOST}_i - \left[ \min(\text{rnd}_d(0.0,1.0) \times \text{WARCOST}_{\text{max}}, \text{WCDISPAR}) \right] \\
\text{WARCOST}_{\text{weaker}} &= \text{WARCOST}_i + \left[ \min(\text{rnd}_d(0.0,1.0) \times \text{WARCOST}_{\text{max}}, \text{WCDISPAR}) \right]
\end{align*}
\]

where \( \text{rnd}_d(0.0, 1.0) \) is a random number generated from a uniform distribution, and \( \text{WCDISPAR} \) is the system-level parameter for weighting power disparities. After paying the war cost, the losing side should pay indemnities to the winners. The calculation of indemnities by each member of the losers’ group is based on the equation (Ibid.):

\[
\text{INDEM}_d = \text{repar}_i \times \text{POW}_d
\]

where \( \text{repar}_i \) means the reparations rate for the state and \( \text{POW}_d \) the power level of the defeated group. On the other hand, each member of the winners’ group will get a part of the total indemnities provided by the losers following the equation (Ibid.):

\[
\text{INDEM}_v = \left( \frac{\text{POW}_v}{\sum_{v=1}^{d} \text{POW}_v} \right) \times \sum_{d=1}^{D} \text{INDEM}_d
\]

91
where $D$ represents the total number of the defeated group, and $V$ the total number of the victorious group. After paying the indemnities, the leader of the defeated group should pay another kind of spoils in the form of territorial cells. The initiator or the target state, if they are in the losers' group, should pay at least one territorial cell to the winner group. If they are multi-cell states, then the number of cells to be transferred will be calculated by multiplying the number of cells to the scope definitions ($LV$ for the initiator, $1.0 - LV$ for the target). The transferred cells will be distributed to the winning group according to their relative power share. No cell is divisible, so that whenever only one cell is to be transferred, the strongest member of the winners' group will take it. After this process, the iteration ends with the recalculation of power level for each country and the natural growth of resources before the next run begins. This recalculation process begins with the summation of the resources A and B at the system-level, and then calculates relative share of each state's resources of A and B. The relative share of each resource and the metabolism rate (for each state) are used again for the calculation of power level.

### 3.3.3 Trade Module

The trade module that is to be added to the security-only world follows the algorithm suggested by Epstein and Axtell's Sugarscape model (1996). In their simulation of society, the trade module is only a part of the big project but it is well-grounded in the microeconomic theory and the bottom-up style CAS philosophy. The Sugarscape model is a general model that can be applied to many social science issues, but one of its promising features is the agent-based modeling. That is, the market in the
Sugarscape model does not assume any perfect information at the individual level nor at the system level. Instead, each trading agent holds only local information and behaves accordingly. Trade happens upon the basis of this kind of “social computation” at the local level (Epstein and Axtell, 1996, 94-96). This contrasts with the neoclassical image of the Walrasian equilibrium, as described by Kreps (1990, 193-195).21

The model of trade used in the Sugarscape (and in this study) does not follow the Walrasian exchange model that assumes a central auctioneer who has all the information on the demand and supply levels in the market. Instead, Epstein and Axtell suggested another trade model based on the assumption of local information, according to their fundamental philosophy of the agent-based model (ABM). In the perspective of ABM (and CAS), the neoclassical Walrasian model of trade is nonrealistic. First of all, the idea that all consumers take the same rate of exchanges is not real. The Walrasian idea is based on the so-called “tatonnement” adjustment process in which no trade is executed until the market clears. This means that consumers (or trading partners) should wait for the agreement of a universal price of the goods they want to consume (or trade). The Walrasian equilibrium assumes a kind of “auctioneer” who adjusts the global equilibrium price or a central authority who examines all the consumers’ demand and set the price upon those demands. These hypothetical mechanisms are unrealistic in that they are intended to achieve the equilibrium of all markets for all consumers at once (Kreps, 1990, 196). Epstein and Axtell, therefore, take the other approach that presupposes the

---

21 The Walrasian equilibrium is based on the assumption of “pure exchange” which is composed of goods, consumers, and utility functions and endowments for each consumer. That is, perfect market equilibrium is possible upon a given level of endowment (Kreps, 1990, 187-189).
mechanism of separate market operation or the dyadic trade relations between two individual traders who have only local information about price.22

In order for an agent to choose trade as her action, we have to assign a certain level of satisfaction to each of two resources that determines the necessity of trade. Epstein and Axtell used the concept of "marginal rate of substitution (MRS)" of one resource to another for a trading agent. In a very simple scheme, if a state's resource A is relatively scarcer than resource B in terms of the agent's metabolism rate, then it will increase its welfare level by trading resource B for resource A with other countries. Upon the calculation of expected welfare (or power) level, the expected utility of trade is determined by MRS, of which the function is (Epstein and Axtell, 1996, 102):

\[
MRS = \frac{dw_2}{dw_1} = \frac{\partial W(w_1, w_2)}{\partial w_1} = \frac{m_1 w_2}{m_2 w_1} = \frac{w_2}{w_1}
\]

where \( w_1 \) and \( w_2 \) are the endowment levels of resource A and B, respectively, and \( m_1 \) and \( m_2 \) are the metabolism levels for resource A and B, respectively. As such, the value of MRS represents the relative internal scarcity of the two resources. Therefore, an agent having MRS less than 1.0 will feel relatively short in its level of resource A, and will trade relatively rich resource B for resource A with the trade partner. The same logic

---

22 This process assumes a step-by-step exchange process of "retrading," which assumes the execution of trade at current prices until all the traders are satisfied. This method is in contrast to the "tatonnement" process mentioned before. Without the income effect, the two processes have the same result, but with the recalculation of power (or welfare) level and resource levels in this study, the consequent effect makes these two processes produce different results.
applies to her trading partner, so that a state will export resource B and import resource A if its MRS is bigger than that of its trading partner. On the other hand, it will export resource A and import resource B if its MRS level is less than that of its trading partner (Epstein and Axtell, 1996, 103). Once an agent decides to trade with another agent, then its price should be decided. The function for price between two these two trading partners is determined by the equation (Epstein and Axtell, 1996, 104):

\[ p(MRS_A, MRS_B) = \sqrt{MRS_A \times MRS_B} \]

Also, once the price is decided, the quantity of trade will be determined. The actual quantity of resources to be traded as follows: if \( p > 1 \), \( p \) units of resource A are exchanged for 1 unit of resource B. If \( p < 1 \), then 1 unit of resource B is exchanged for 1/\( p \) units of resource A. Once a trade happens between two agents, it goes on unit by unit until each side’s MRS does not crossover (Epstein and Axtell, 1996, 104). The overall flow of the trading process is described in Figure 3.6. Also, Figure 3.7 shows the combined diagram of algorithms devised for the simulation of this study.

3.4 Variables and Equations

The variables of this simulation study will be organized differently according to the propositions in a way that is appropriate to each topic, so they will be discussed separately in each chapter. In particular, the group of five control variables – standard
deviation in power distribution, standard deviation in error ratio distribution, maximum war cost parameter, reparation parameter, disproportionate war cost – plus extra variables that Cusack and Stoll used (for testing supplementary propositions) will be explained in the next chapter on the basic run. The variables that are not introduced in the basic run will be discussed in detail in each chapter where those are used. The structure of equations, according to each proposition, will also be organized and discussed in the research design section of each chapter. In the following chapters, the structure of overall simulation runs and the iteration format (how many iterations per simulation run are designed) also will be introduced before the main discussion.
Figure 3.6: Epstein and Axtell's Trade Algorithm
Choose the neighbor that makes the biggest EU, \( \text{war} \)

War algorithm (Cusack and Stoll)

Choose the neighbor that makes the biggest EU, \( \text{trade} \)

Trade algorithm (Epstein and Axtell)

Adjust power

End of iteration

End of run

Iteration limit?

YES

NO

Is any initiator selected?

YES

NO

EU, \( \text{war} \) > EU, \( \text{trade} \)?

Select an actor \( i \) (based on power level and EU's)

Calculate EU, \( \text{war} \), EU, \( \text{trade} \) on neighbors

Update power (Start iteration)

Initialize (Start run)

End of iteration

End of run

Figure 3.7: Combined Algorithm
CHAPTER 4

TRADE AND WAR

4.1 Research Themes

This chapter deals with the first three propositions, based on Puzzle 1 – “How do the micro-level properties of individual agents affect the systemic structures of world politics?” In particular, I would like to see how the introduction of the trade option at the micro-level (the state agent level) changes the systemic phenomena such as polarity and power concentration. Actually, this chapter starts from the replication of the base run of Cusack and Stoll’s original model (1990) of the security-only world or the Realpolitik world. While the main theme of this chapter investigates the impact of the change in state agents’ property on the structure of world politics, I would like to follow their basic scheme in order to compare the differences and similarities of the two worlds – the security-only world and the security-and-trade world – along the main control variables. So the title of this chapter – trade and war – implies the two most important, contrasting elements in modeling the interactions among state agents.
Cusack and Stoll, in their original model, built an "automatic stabilization model" in which the individual states' behaviors generate specific systemic phenomena—especially the stability of world politics. Their idea was based on Bremer and Mihalka's description of the realist puzzle—"How is stability achieved in multistate systems?" In the old model by Bremer and Mihalka, stability is understood as a low level of violent conflicts in a balance-of-power or the emergence of an empire (that controls the whole world). In particular, Cusack and Stoll were interested in simulating the "automatic" version among the three models of international stability suggested by Inis Claude—the automatic process, the semiautomatic process, and the manually operated process (Bremer and Mihalka, 1977, 304). The automatic process of stability does not assume any intentional actor who tries to balance or contain others at the global level, like Britain in the 19th century. Every actor follows a set of rules in this process without any controller, but the stability of world politics is possible "automatically," unlike the semiautomatic process or the manually operated process.

In order to implement the automatic version of system stability—like Adam Smith's "invisible hand"—Cusack and Stoll examined three dependent variables: system endurance, balance-of-power, and state survival. I would like to use the same set of dependent variables in my replication for comparison. But the basic purpose of this chapter is observing the systemic effects of the change in individual states' properties—represented by the addition of the "trade" option in their interactions with neighbor countries. As Bremer and Mihalka mentioned, this kind of work is to observe the "logical consequences of states acting according to a particular set of decision rules" rather than to replicate the real-world observations in an empirical sense (1977, 306).
As such, this chapter will investigate the propositions suggested in the previous chapter – from Proposition 1 to Proposition 3 as follows:

**Proposition 1:** With the introduction of trade, the chance of the system of states to endure rather than collapse into universal empire will be higher than in a world where only security relations (including alliance formation) are possible.

**Proposition 2:** With the introduction of trade, the probability of balance-of-power will be higher than that of the security-only world.

**Proposition 3:** With the introduction of trade, the survival chance of states within systems will be higher than that of the security-only world.

These three propositions are generated from Puzzle 1 and reorganized according to the basic scheme of Cusack and Stoll’s automatic stabilization model. Therefore, the same set of explanatory variables as well as three main dependent variables will be used in this chapter, except for some modifications that are necessary for methodological purposes. For each proposition, I will reproduce two artificial worlds, one as the security-only world (the basic replication of Cusack and Stoll’s original model) and the security-and-trade world. Comparisons will be made for each proposition between the two worlds, and discussions will follow based on the observations of simulation results.
4.2 Research Design

4.2.1 Variables

Cusack and Stoll used eight variables to explore—five as major factors and extra
three as experimental factors. Table 4.1 is the list of variables that they used for their
original simulations. Five major variables are power estimation error, power distribution,
war cost maximum, reparations, and LV sigma. The first two variables control the
parameters by fixing the standard deviation of error and power distributions at the
beginning of each run, respectively. That is, the values for power estimation error and
power distribution are assigned to each run through generating random numbers within
the range set with these standard deviations. The others directly control the value of
systemic parameters. For example, at every run, all the state agents should hold the same
war cost rate, reparations rate, and LV sigma by the end of iteration. These parameters
are distributed each run by extracting random numbers from the normal distribution. On
the other hand, Cusack and Stoll used three experimental variables that also represent
some systemic parameters that took their attention—variances in power growth rate,
disproportionate war cost, and the parameter for war ties—but they were used in a limited
set of runs rather than throughout the whole series of runs. Table 4.1 also shows the list
of these extra factors as well as the main five factors, along with the possible ranges of
each factor. Notice that the values of factors are set as “discrete” (i.e., low, medium, or
high, despite the numbers assigned to those labels) rather than “continuous” which I will
discuss later.
<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
<th>Value Ranges</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Major Variables</strong></td>
<td></td>
<td>Low</td>
</tr>
<tr>
<td>Power Distribution</td>
<td>Standard deviation for the distribution of initial power scores assigned to states in system (average state power equals 1.0 at initial iteration).</td>
<td>0.1</td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>Standard deviation for the distribution of the power estimation error assigned to states in system (average accuracy score equals 100%).</td>
<td>0.1</td>
</tr>
<tr>
<td>War Cost Max</td>
<td>Maximum proportion of a state’s power that is lost through participation in war.</td>
<td>0.05</td>
</tr>
<tr>
<td>Reparations</td>
<td>Proportion of a losing state’s power that is given over to the winning side in a war.</td>
<td>0.1</td>
</tr>
<tr>
<td>LV Sigma</td>
<td>Value for the LV (likelihood of victory) function parameter that controls the relationship between the power ratio (the initiator’s coalition relative to the target’s) and the probability of victory for the initiator.</td>
<td>1</td>
</tr>
<tr>
<td><strong>Experimental Variables</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variance in Growth Rate of Power</td>
<td>Standard deviation for distribution of internal power growth rates across states in system. Values used: 0, .005 (average growth rate equals .01 per iteration).</td>
<td>0.0</td>
</tr>
<tr>
<td>Disproportionate War Costs</td>
<td>Value for relationship between prewar power ratio of stronger to weaker side and asymmetric war costs (stronger side suffers smaller relative costs than weaker side). Values used: 0, ½ of the maximum war cost parameter.</td>
<td>0.0</td>
</tr>
<tr>
<td>War Ties (Indecisive War)</td>
<td>Value for exponent of function determining whether a war ends in a tie. Value used: 0 (no ties), 3 (high variant), 5 (low variant).</td>
<td>0</td>
</tr>
</tbody>
</table>

(Source: Cusack and Stoll, 1990, 106)

Table 4.1: Cusack and Stoll’s Variable Set
While this study follows the basic scheme of this variable set, I have to modify some of them for the purpose of my research. First of all, due to the addition of the “trade” option, I have to assume the decision-making style of each state agent is a rational actor who maximizes expected utility of her choices, while Cusack and Stoll used the “primitive power seeker” type as their basic actor in simulations. Including the trading option, for my research, needs another assumption that state agents “calculate” the benefits for their choices. This study is not for the comparison between different decision-making styles – like Cusack and Stoll’s comparison (1990, Chapter 5) between primitive power seeker, balancer-of-power, collective security seeker, and rational utility maximizer – but a comparison between the world composed of agents who have security-or-not options, and the world composed of agents who have security-and-trade options. Any consistent comparison between these two schemes should assume the same type of agents, but it is not possible if I fix the agent style as “primitive power seekers” because the state agents have no common criterion for choosing between war and trade in that case. So I have to assume that the state agents are “utility maximizers” in their decision-making. As such, the research design of this study is limited by the type of state agents.

Second, I have to change or eliminate some factors from the variable list and add another for the purpose of this research. The first element to change is the “LV sigma” variable that determines the likelihood of victory parameters for each state. This factor is important in its effect on the relationship between the power ratio (between two states on a dyad) and the likelihood of victory (See Figure 3.5). The reason for elimination is that this “not only controls the actual outcomes of wars but is also used by the highly rational states to assess their chances of winning or losing a potential war” (Cusack and Stoll, 104).
1990, 181). That is, the agents of utility maximizers have to calculate their expected utility of war, which needs the parameter value of LV sigma for the winning probability and the consequent expected benefits of war. At the same time, it determines the result of any war. For this reason, I have to fix its value to 1.0, rather than the varying scheme between 1.0 and 5.0 which was used by Cusack and Stoll for the primitive power seekers. Another factor that is to be eliminated is the “war ties” parameter which is not included in this study for the purpose of simple simulation implementation. That is, like the civil war module, the tied-war scheme does not have priority over other factors and modules in investigating the difference between the security-only world and the security-and-trade world; so that every war should end with a winner and a loser. The parameter for the variance in growth rate of power is included in the simulations, but it is not in the analysis of data because it was not clearly analyzed even in Cusack and Stoll’s automatic stabilization model. On the other hand, I would like to use another parameter that controls the variance in the distribution of metabolism rates, which is related to the internal scarcity of resources, the marginal rate of substitution, and consequently the expected utility of trade.

Third, I would make the format of variations for each parameter different from Cusack and Stoll’s so as to reflect the basic idea of the agent-based modeling. That is, I would like to reduce the number of system-level parameters as far as I can, while

---

23 According to Cusack and Stoll, if we use this parameter in simulations, it is possible that “any highly rational state would never anticipate a positive expected utility if the parameter controlling the shape of the curve generated by the function took on a value greater than 1.0” (Cusack and Stoll, 1990, 181). For this reason, they also fixed the parameter of LV sigma to 1.0 especially for the case of rational utility maximizers (Ibid., 162).

24 Bremer and Mihalka’s old scheme does not include the tied-war situations, and Cusack and Stoll’s war-ties parameter was applied only to some limited situations in which the war-ties function is determined by the likelihood of victory function. For the function of war-ties, see Cusack and Stoll (1990, 86).
converting them to individual-level ones. In the case of Cusack and Stoll's original model, parameters were fixed to certain values at the initial stage of each run. This has two problems. The first problem is that the values of parameters are equally applied to each state in the same run. For example, when they set the reparations rate to 0.1 for a run, then every state at different iterations (in the same run) has the same level of reparations. This is unrealistic, and it contradicts to the basic bottom-up philosophy of the CAS in which the actors (individual agents) are influenced more by endogenous factors rather than by the global, systemic factors. The second problem is that Cusack and Stoll set the values of those systemic variables as discrete. Thus, after they ran different sessions of simulations with different parameters – such as the reparations rates of 0.1, 0.2, and 0.3 – they compared the influences of this factor on the dependent variable across these three pre-assigned values. This kind of treatment seems to have reflected the cost of simulation runs at that time (and maybe some other conditions), but it needs not be so in this period of high-level computing machines.

Therefore, I would like to change this format considering the two problems discussed above. First of all, the major factors – reparations rate, maximum war cost, and disproportionate war cost – will be converted to measure the individual-level parameters rather than the system-level parameters. For example, reparations rate will be assigned to each individual agent state differently (of course, with a random number from the normal distribution) whenever the parameter value is necessary. On the other hand, the other two system-level parameters – the standard deviation of power distribution and the standard deviation of error distribution – will be used as those of Cusack and Stoll’s original design, because these are the parameters for the distributions rather than for individual
agents' properties. For the second problem of discrete variation scheme, I would like to make the values of variables vary on a continuous spectrum. The standard deviations (S.D.) and other parameters will be assigned random numbers generated from the uniform distribution (S.D. for power distribution and for error distribution) or from the normal distribution (war cost maximum, reparations, and disproportionate war cost). These variables can take any values on the spectrum within the ranges, respectively, rather than the discrete values predetermined before the experiment.

As such, this study tries some modifications to Cusack and Stoll's original simulation model in terms of variable composition and their properties. The changed composition of variables is specifically only for this chapter because of the purpose of comparison, while different designs are to be expected in other chapters according to each research topic. The basic dependent variable set — system endurance (and multiplicity), balance-of-power, and state survival — is the same as that of Cusack and Stoll, even though a slight modification is expected for special tests. Table 4.2 shows what variables are modified and what are included or excluded in this study vis-à-vis Cusack and Stoll's original model. There are a couple of other variables that are specific to each series, but they will be discussed whenever appropriate. As such, Table 4.2 represents only the basic set of variables used in the simulation runs throughout this study, from Chapter 4 to Chapter 6.

25 For example, a standard deviation 0.27 is assigned to the error distribution of a specific run from a uniform random number between predefined range between 0.1 and 0.5. Then each individual state agent will be assigned a random value of errors whenever she estimates power from the normal distribution with mean 0.0 and standard deviation 0.27. This value of standard deviation is used to the end of the run.
Table 4.2: Revised Set of Variables

Table 4.2: Revised Set of Variables

<table>
<thead>
<tr>
<th>Variables</th>
<th>Modifications</th>
<th>Included in Analysis?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Distribution</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>War Cost Max</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Reparations</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>LV Sigma</td>
<td>Fixed to 1.0</td>
<td>No</td>
</tr>
<tr>
<td>Variance in Growth Rate of Power</td>
<td>Included in simulations, excluded in analysis</td>
<td>No</td>
</tr>
<tr>
<td>Disproportionate War Costs</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>War Ties</td>
<td>Excluded</td>
<td>No</td>
</tr>
<tr>
<td>Variance in Distribution of Metabolism Rate</td>
<td>Newly generated</td>
<td>Yes</td>
</tr>
</tbody>
</table>

4.2.2 Equations

For the three proposition tests, several equations are suggested here. As explained, Proposition 1 deals with the impact on system endurance (the dependent variable) by the different sets of individual agent’s choices – without trade and with trade:

**Proposition 1:** With the introduction of trade, the chance of the system of states to endure rather than collapse into universal empire will be higher than in a world where only security relations (including alliance formation) are possible.
For the two situations with the security-only world and the security-and-trade world, two dependent variables will be used – whether the system collapses into an empire and the duration of the runs until the system collapses into an empire – to measure the “multiplicity” of the system. Both dependent variables are used in Cusack and Stoll’s original scheme; the first variable is a binary one about whether the system collapses into an empire before the run reaches iteration 1,000 (or 2,000 or 3,000 whenever necessary for comparison), and the second is a continuous variable measuring the length of time for a system to collapse into an empire. While the concept of “stability” was used in Bremer and Mihalka’s old scheme for the automatic stabilization of world politics, it seems ambiguous what it really means. Actually, Bremer and Mihalka introduced two interpretations for this concept – stability as a “low level of violent conflict” and a “small number of states and the distribution of power that is relatively constant” (Bremer and Mihalka, 1977, 304). On the other hand, Cusack and Stoll used an unambiguous concept “plurality” of the system for their dependent variable of system endurance (Cusack and Stoll, 1990, 107). Considering this confusion, this study uses the concept of “multiplicity” (rather than “stability”) for the clear measurement of dependent variable, following Cusack and Stoll. Two equations, therefore, will be:

\[
\text{Multiplicity} = \beta_0 + \beta_1 \cdot \text{pow}_sd + \beta_2 \cdot \text{error}_sd + \beta_3 \cdot \text{warcost} \\
+ \beta_4 \cdot \text{wcdispar} + \beta_5 \cdot \text{repar} + \beta_6 \cdot \text{meta}_sd
\]

\[
\text{Duration of runs} = \beta_0 + \beta_1 \cdot \text{pow}_sd + \beta_2 \cdot \text{error}_sd + \beta_3 \cdot \text{warcost} \\
+ \beta_4 \cdot \text{wcdispar} + \beta_5 \cdot \text{repar} + \beta_6 \cdot \text{meta}_sd
\]

(where \(\text{pow}_sd\) is the standard deviation of power distribution, \(\text{error}_sd\) is the standard deviation of power estimation error,
warcost is the maximum proportion of power to lose in a war, \textit{wcdispar} is the prewar power ratio to asymmetric war cost, \textit{repar} is the proportion of the loser’s power given to winners, and \textit{meta\_sd} is the standard deviation of metabolism distribution.

As the first equation includes a binary dependent variable, probit will be used for evaluating this proposition. On the other hand, the dependent variable (the number of iterations a run persists) of the second equation is a truncated continuous variable, so a tobit analysis will be used with its lower limit 97 and upper limit 1,000. Total runs of simulations for these equations will be 10,000 that might be an enough number of cases to investigate diverse combinations of parameters that are generated by random number generators.

Proposition 2 is about the impact of state property change (without and with trade) on the balance-of-power pattern at the systemic level. As discussed before, balance-of-power is a characteristic pattern that emerges (automatically in many cases) in world politics. Several mechanisms have been presented to explain the working of balance-of-power — an automatic system, a semi-automatic system, and a manual system — but this study focuses on the working of the automatic balance-of-power system which was a research target of Cusack and Stoll (also of Bremer and Mihalka). Programming

\footnote{As the number of states is 98 at the initial stage, the minimum number of iterations is 97 for the collapse of the system into an empire. On the other hand, as each separate run will end at iteration 1,000, the upper limit for the tobit model is set to 1,000. Whenever necessary, I will run the same series with 2,000- and 3,000-iteration options for comparison; in these cases, the upper limit for tobit model will be set to 2,000 and 3,000, respectively.}

\footnote{Cusack and Stoll’s experiment is composed of 8,748 runs of simulations, which is composed of 2,916 combinations of eight (discrete) parameter values multiplied by three random number seeds (Cusack and Stoll, 1990, 107). My design is composed of 10,000 runs which does not fix the parameters at specific values but lets them be generated while running simulations. Technically, this style of programming is called “dynamic-binding” which is one of major features of the object-oriented programming.}
for each state’s utility-maximization mechanism in deciding war, trade, and alliance is necessary, but not for any balance-of-power mechanism at the system-level:

Proposition 2: With the introduction of trade, the probability of balance-of-power will be higher than that of the security-only world.

Here, I would like to see whether there is any increase in the balance-of-power patterns measured by two concepts – deterrent balance-of-power and defensive balance-of-power – which will be discussed below, if I allow state agents to choose trade as well as war with neighbor states (at the individual level). With the trade option introduced, it is natural for us to expect a lower frequency of war; but we may not easily predict how much the balance-of-power tendency will increase (or decrease) once wars happen in the system. Thus Proposition 2 hypothesizes about a difference in the balance-of-power patterns between the two artificial worlds.

I will look at the pattern of balance-of-power using Cusack and Stoll’s two concepts – deterrent balance-of-power and defensive balance-of-power. According to them, the “deterrent balance-of-power” is states’ alliance that stops the escalation to war, while the “defensive balance-of-power” accompanies the change in the power ratio favorable to the target or the defensive side once a war occurs (1990, 120). They measured the impact of the explanatory variable group on the patterns of balance-of-power measured by these two criteria. For the explanatory variable group, they added the iteration group variable, upon the assumption that the difference in iteration stages will have different impacts on the balance-of-power pattern at the system-level. I will use all
of these explanatory variables except the LV sigma, which is eliminated from the
research design of this study due to its irrelevance in terms of the agent’s property.

Like Cusack and Stoll’s scheme, I will code the case as “deterrent balance-of-
power” if a war is intended with the selection of a target by an initiator but it does not
occur due to the formation of a defensive alliance. Also, I would code the case as
“defensive balance-of-power” if a war happens but with the power ratio of the defensive
alliance to the offensive alliance becomes higher than the original initiator-target power
ratio due to the formation of a defensive alliance. In other words, deterrent balance-of-
power refers to the case when a war is deterred by the formation of a defensive alliance
while defensive balance-of-power represents the case when a war happens but the
defensive part gets advantages in power against the initiator and/or the offensive alliance.
One point here is that I would not include some cases that are not relevant to treat as
“balance-of-power.” Cusack and Stoll regarded the case as “defensive balance-of-power”
if an initiator chooses a target for war and a war happens anyway. However, if we look
closely at the process of war initiation and its escalation through alliance formation, we
know there are some problems in that scheme. For example, a target state may not find
any alliance candidate or she may not receive any response for her bid for alliance after
the initiator starts her war intention. These cases should not be regarded as balance-of-
power because no alliance is formed before a war occurs. Table 4.3 shows the overall
process of alliance formation.
<table>
<thead>
<tr>
<th>Cases</th>
<th>War Target Selected</th>
<th>Defensive Alliance Formed with EU &lt; 0.0</th>
<th>Defensive Alliance Formed with EU &gt; 0.0</th>
<th>Offensive Alliance Formed with EU &lt; 0.0</th>
<th>Offensive Alliance Formed with EU &gt; 0.0</th>
<th>Defensive Alliance's EU &gt; 0.0 (without Enhanced Alliance)</th>
<th>Enhanced Defensive Alliance, But Not Increased</th>
<th>Enhanced Defensive Alliance, Increased</th>
<th>War Happened</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 4.3: Process of Alliance Formation
In Table 4.3, all the steps from the selection of war target by the initiator to the occurrence of war are described in detail. Running the iterations of simulation will generate cases that match one of the steps, from Case 1 to Case 8, according to the success/failure of alliance formation. As Case 3 and Case 4 do not lead to the occurrence of war after the formation of defensive alliance, they are regarded as the cases of “deterrent balance-of-power,” both in my scheme and in Cusack and Stoll’s scheme. Case 5 to Case 8 represent the “defensive balance-of-power” only when they accompany a change in the offensive-defensive alliance power ratio. As such, the setting of deterrent and defensive balance-of-power behaviors is the same between this study and Cusack and Stoll’s, as far as each case falls into Case 3 to Case 8. However, Case 1 and Case 2 do not represent any alliance that is successfully established to balance the offensive initiator, so that they should not be coded as “balance-of-power.” Therefore, while Cusack and Stoll included Case 1 and Case 2 in their selection of balance-of-power cases, I will not do so for the reasons discussed so far.

For both sets of worlds – the security-only world and the security-and-trade world – the following two equations will be estimated and compared with each other. Here, the variable “iteration group” is included in the equation as it is assumed that different time periods will differently affect the balance-of-power patterns (Cusack and Stoll, 1990, 121). On the other hand, for the analysis of defensive balance-of-power, another explanatory variable “initial power ratio” is added in order to see the impact of initial configuration of power. Initial power is measured by the relative share a state holds at the initial stage of simulation runs, which is generated by a random number generator with average 1.0 and standard deviation which is also randomly assigned within the pre-

114
determined range. Particularly, the analysis of the defensive balance-of-power pattern examines only the cases that ended in war, in order to see the influence of the explanatory variable set on the change in the power ratio between the initiator and the target (and their alliances, respectively). Samples will be collected from 100 random runs of simulations for the first equation, while another set of samples for defensive balance-of-power will be collected from the first set for the second equation.28

\[
\text{Deterrent balance-of-power} = \beta_0 + \beta_1 \cdot i\_group + \beta_2 \cdot \text{pow}\_sd + \beta_3 \cdot \text{error}\_sd \\
+ \beta_4 \cdot \text{warcost} + \beta_5 \cdot \text{repar} + \beta_6 \cdot \text{meta}\_sd
\]

\[
\text{Defensive balance-of-power} = \beta_0 + \beta_1 \cdot i\_group + \beta_2 \cdot \text{it}\_ratio + \beta_3 \cdot \text{pow}\_sd \\
+ \beta_4 \cdot \text{error}\_sd + \beta_5 \cdot \text{warcost} + \beta_6 \cdot \text{repar} + \beta_7 \cdot \text{meta}\_sd
\]

(where \(i\_group\) is the collapsed criteria of time period from 1 to 1,000, \(\text{it}\_ratio\) is the power ratio between the initiator and the target, \(\text{pow}\_sd\) is the standard deviation of power distribution, \(\text{error}\_sd\) is the standard deviation of power estimation error, \(\text{warcost}\) is the maximum proportion of power to lose in a war, \(\text{repar}\) is the proportion of the loser's power given to winners, and \(\text{meta}\_sd\) is the standard deviation of metabolism distribution)

While Proposition 1 and 2 are explicitly focusing on the systemic patterns – system multiplicity and balance-of-power – affected by the change of state properties, Proposition 3 deals with state survival rates. There are two reasons for the setting of this proposition: first, it is intended to compare this study with Cusack and Stoll’s original study side-by-side, according to the configuration of dependent variables; second, it deals

---

28 Cusack and Stoll also did 100 random runs to generate 38,497 cases in which the initiator selected the target for war. About 43 percent of the total iterations escalated to war in their simulations (Cusack and Stoll, 1990, 121). In a comparative implementation of simulation, the security-only world in this study is expected to generate more balance-of-power cases than the security-and-trade world, because the latter allows trade option as well as war.
with the state survival level not at the unit-level but at the system-level. That is, it interprets the survival rate in a statistical sense; it need not be applied to specific individual states. In this sense, Proposition 3 can match the previous ones, in terms of its relation to Puzzle 1.

**Proposition 3:** With the introduction of trade, the survival chances of states within systems will be higher than that of the security-only world.

Cusack and Stoll used three dependent variables for testing the impact on the state survival rate – survival to iteration 100, survival to iteration 1,000, and the number of iterations survived for each initial state. The explanatory variable set is divided into two groups – one is state-level variables, and the other is system-level variables. State-level variables include the share of system power at entry into the system, error rates in estimating its own and others' power, and the geographic position. System-level variables include power distribution, power estimation error, reparations, LV sigma (which is not included in this study), and war cost maximum. Also Cusack and Stoll measured state survival rate to iteration 200 with the introduction of an immediate environmental variable, which is measured by the combined power of neighbor states. The following equations are based on the collection of cases that survived to iteration 100

---

29 The index of geographic position \( (GP_i) \) is constructed by using the column \( (C_i) \) by row \( (R_i) \) positions of a state, according to Cusack and Stoll. The core cells (cell 49 and 50) in the world system have \( GP_{49} = 1 \), while the outer cells at the edges have \( GP_{28} = 28 \). The following rules determine each cell’s \( (i.e., \) each initial state’s \( ) \) \( GP_i \): (1) \( C_i > 7 \Rightarrow GC_i = C_i - 7 \); (2) \( C_i < 8 \Rightarrow GC_i = |C_i - 8| \); (3) \( R_i > 4 \Rightarrow GR_i = R_i - 3 \); (4) \( R_i < 5 \Rightarrow GR_i = |R_i - 5| \); (5) \( GP_i = GC_i; GR_i \) (Cusack and Stoll, 1990, 135).
and that are expected to survive to iteration 200 or to die before iteration 200.\textsuperscript{30} I will replicate these variable sets with some modifications like the previous sections – i.e., eliminating the LV sigma variable and adding the standard deviation of metabolism distribution. The sample will be extracted from 1,000 runs of simulations, which will generate a total of 98,000 individual cases.\textsuperscript{31} Thus the equations for Proposition 3 are:

\[
\text{Survival to iteration 100} = \beta_0 + \beta_1 \cdot \text{rel_pow} + \beta_2 \cdot \text{err_own} + \beta_3 \cdot \text{err_oth} \\
+ \beta_4 \cdot \text{geo} + \beta_5 \cdot \text{pow_sd} + \beta_6 \cdot \text{error_sd} + \beta_7 \cdot \text{repar} \\
+ \beta_8 \cdot \text{warcost} + \beta_9 \cdot \text{meta_sd}
\]

\[
\text{Survival to iteration 1,000} = \beta_0 + \beta_1 \cdot \text{rel_pow} + \beta_2 \cdot \text{err_own} + \beta_3 \cdot \text{err_oth} \\
+ \beta_4 \cdot \text{geo} + \beta_5 \cdot \text{pow_sd} + \beta_6 \cdot \text{error_sd} + \beta_7 \cdot \text{repar} \\
+ \beta_8 \cdot \text{warcost} + \beta_9 \cdot \text{meta_sd}
\]

\[
\text{Number of iterations survived} = \beta_0 + \beta_1 \cdot \text{rel_pow} + \beta_2 \cdot \text{err_own} + \beta_3 \cdot \text{err_oth} \\
+ \beta_4 \cdot \text{geo} + \beta_5 \cdot \text{pow_sd} + \beta_6 \cdot \text{error_sd} + \beta_7 \cdot \text{repar} \\
+ \beta_8 \cdot \text{warcost} + \beta_9 \cdot \text{meta_sd}
\]

\[
\text{Survival to iteration 200} = \beta_0 + \beta_1 \cdot \text{rel_pow} + \beta_2 \cdot \text{err_own} + \beta_3 \cdot \text{err_oth} \\
+ \beta_4 \cdot \text{geo} + \beta_5 \cdot \text{pow_sd} + \beta_6 \cdot \text{error_sd} + \beta_7 \cdot \text{repar} \\
+ \beta_8 \cdot \text{warcost} + \beta_9 \cdot \text{neighpow}
\]

(where \text{rel_pow} is the share of system power at entry into system, 
\text{err_own} is the error rate in estimating own power, 
\text{err_oth} is the error rate in estimating others' power, 
\text{geo} is the level of geographic position, 
\text{pow_sd} is the standard deviation of power distribution, 
\text{error_sd} is the standard deviation of power estimation error, 
\text{repar} is the proportion of a losing state's power given to winners, 
\text{warcost} is the maximum proportion of power to lose in a war, 
\text{meta_sd} is the standard deviation of metabolism distribution, and 
\text{neighpow} is the combined power of neighboring states)

\textsuperscript{30} According to Cusack and Stoll, a segment of 100 iterations (between iteration 100 and iteration 200) is “long enough to allow neighborhood effects to appear but short enough to prevent the local environment of states from altering so radically that a static measurement of it becomes completely invalid” (1990, 131).

\textsuperscript{31} Cusack and Stoll ran 20 runs for this analysis, which generated only 1,960 cases of initial states to be analyzed (\textit{ibid.}, 125).
4.3 Analysis of Results

4.3.1 System Endurance

The patterns of system endurance between the security-only world and the security-and-trade world show a big difference between them, as expected. Figure 4.1 shows the distribution of runs in the security-only world. In this world, all the runs ended with the emergence of an empire and with the disappearance of system multiplicity before iteration 200. States in this world calculate their benefits and costs of probable war against each of their neighbors (without any option for trade), and then choose a target that has the biggest positive expected utility. If a state does not have any positive expected utility of war against all of her neighbors, she will not be considered as a candidate for the initiator. The result of simulation runs shows the brutal nature of world politics. Actually, this is a very similar result as that of Bremer and Mihalka’s, but a little bit different from Cusack and Stoll’s. Bremer and Mihalka’s *Machiavelli in Machina* resulted in the emergence of an empire before iteration 300, even though the decision-making style of it is primitive power-seekers unlike the rational utility maximizer in this study (1977, 326).

On the other hand, in Cusack and Stoll’s original study, simulation runs ended with empires in only about 50 percent of cases (1990, 114). There are many reasons for this difference, like the removal of civil war module and war-ties scheme in my study, but the most probable cause is due to fixing LV sigma to 1.0, as discussed before. If I use the LV sigma value that varies between predetermined ranges, as shown in Figure 4.2,
the result shows a very similar pattern as that of Cusack and Stoll's. In this sense, the
result with the security-only world (with fixed LV sigma) will not be directly compared
to that of the security-and-trade world (with varying LV sigma). Only a simple
comparison will be made between the security-only world with its LV sigma value
varying and the security-only world, and between the security-and-trade worlds with
different iteration runs. From Figure 4.1 to Figure 4.5, we can observe overall distribution
patterns of conflict in different worlds. All figures are related to Proposition 1.

![Figure 4.1: System Endurance (Security-Only World, LV Fixed to 1.0)](image)

32 In the real runs of the simulation, the impact of LV sigma is much bigger than other factors in this study
as well as in Cusack and Stoll's. As the value of LV sigma gets bigger, as shown in Figure 3.5, the
likelihood of victory relies more on chance factors than on the pure level of power difference. While
acknowledging this kind of chance factors, it produces some annoying results that are hard to explain. For
example, a very small state can win a war against a very big state, due to the very high LV sigma value (up
to 5.0). In this sense, elimination of the LV sigma scheme (for other reason – i.e., for the characteristic of
the agent's decision-making style) has an advantage in producing a more plausible simulation result.
* This graph shows the difference in the final result if we make the likelihood of victory (LV) sigma value vary rather than fixed. Compare this graph only to Figure 4.1.

**Figure 4.2: System Endurance (Security-Only World, LV Varying)**

**Figure 4.3: System Endurance (Security-and-Trade World with 1,000 Iterations)**
* This and the next graphs need not to be directly related to Propositions. These are just for the visual comparison with the 1,000-iteration setting in Figure 4.3. I’d like to show that nothing is changed even if we extend the iteration limit.

Figure 4.4: System Endurance (Security-and-Trade World with 2,000 Iterations)

Figure 4.5: System Endurance (Security-and-Trade World with 3,000 Iterations)
Figure 4.1 and Figure 4.2 are the distributions of runs in the security-only world, with LV sigma value constant and varying, respectively. With LV sigma varying scheme in Figure 4.2, the distribution of ending rounds gets smoother with a tall column at the far right (iteration 1,000) which might be lowered if we extend the iteration limit. Once we insert the trade module (with LV fixed to 1.0) in the security-only world, we observe a different pattern with more multiplicity in the composition of system agents. Figure 4.3 is the distribution of runs with the trade option (maximum 1,000 iterations). About 10 percent of cases ended with multiple states by the end of iteration limit, while other cases are spread around iteration 500. We can see that the trade option extends the system endurance up to several hundreds of iterations more. Also we can find that the trade option assures the multiplicity of the system, with the inverted-U shape of distribution.

Figure 4.4 and Figure 4.5 are extensions of the same format only with different iteration limits – up to 2,000 and 3,000 iterations. These were experimented separately in order to see whether a loose iteration limit makes the impact of trade different. These figures show that the basic patterns are the same as that of 1,000-iteration version, only with the effect of lowering the crowded column at the end of iterations. This indicates that the mechanism of trade leads to a more multiple system than the security-only world does, but its multiplicity also fades away as time passes anyway. The only difference is the time span in which multiplicity exists. Interestingly, the collapse of multiplicity happens mostly around iteration 500, even though we extend the iteration limit twice and three-times. The consequent bell-shaped pattern of system endurance implies that the combination of the security-and-trade world and its parameters (within ranges) generate this kind of specific multiplicity pattern regardless of iteration limits.
Table 4.4: Probit Analysis of System Multiplicity

As mentioned before, all the simulation runs for the security-only world ended with an empire, so that it is not possible to compare it directly with the security-and-trade world in terms of the binary variable of system multiplicity. Actually, with the LV sigma fixed at 1.0, there is no variation in the dependent variable. So Table 4.4 compares the security-and-trade world and the security-only world with LV sigma value that varies, even though they do not directly match with each other. The impact of trade seems to be great in the effect of the power estimation error, war cost maximum, and disproportionate war cost. The first two factors show the change of directions in their influence, too. The rational actor responds more sensitively and sometimes in a reverse way to these cost-related factors in the security-and-trade world. However, the impacts of power distribution, reparations, and metabolism distribution are not as great in the security-and-
trade world as in the security-only world. This implies that the system composed of utility-maximizing actors assumed in this study is not much influenced by the factors of power distribution, reparations, and metabolism distribution in its maintenance of multiplicity or its endurance before collapsing into an empire.

Now let's see the result of the tobit analysis in Table 4.5. This table includes the LV fixed setting (the first column) again because we can make the dependent variable – i.e., the number of iterations the system keeps multiplicity – vary in the tobit equation. With the introduction of trade, we observe the increase of every factor's impact vis-à-vis that of the security-only world, particularly with a high level of significance. The first and second columns show the coefficients of each explanatory variable of the two worlds, respectively. We can observe that there are big differences in the coefficient values between these two worlds. The last column is the result of t-test after running a tobit analysis for the pooled data. Across the six major explanatory variables, the impact of group (i.e., with the trade option) is tested. The null hypothesis for each variable is that the extra impact from the group difference is zero, for each variable. In the table, we observe that the changes in the impacts by major control factors are significant, except the case of metabolism. Thus we can conclude that the introduction of trade makes the impacts of major variables bigger (with the exception of metabolism rate).

33 The pseudo R² values in this result (0.01 and 0.13 for the security-only world and the security-and-trade world, respectively) is not big, as in Cusack and Stoll's basic run with the primitive power-seekers, which resulted in 0.30 (1990, 109). The low level of pseudo R² for the security-only world (0.01) against other formats (more than 0.10) seems to be caused by the elimination of the LV sigma variation that has recorded the greatest impact on the dependent variable in Cusack and Stoll's original simulation.
Table 4.5: Tobit Analysis of the Number of Iterations

<table>
<thead>
<tr>
<th>Variables</th>
<th>Security-Only World (LV σ = 1.0, Iteration = 1,000)</th>
<th>Security-and-Trade World (LV σ = 1.0, Iteration = 1,000)</th>
<th>H0: Extra Impact = 0.0 F(1,19988)*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>-1.40</td>
<td>194.17 ***</td>
<td>279.37 ***</td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>57.24 ***</td>
<td>1,531.52 ***</td>
<td>2,472.14 ***</td>
</tr>
<tr>
<td>War Cost Maximum</td>
<td>-187.00 ***</td>
<td>-1,734.85 ***</td>
<td>1,518.58 ***</td>
</tr>
<tr>
<td>Disproportionate War Cost</td>
<td>18.18 ***</td>
<td>-421.86 ***</td>
<td>1,350.83 ***</td>
</tr>
<tr>
<td>Reparations</td>
<td>30.13 ***</td>
<td>117.36 ***</td>
<td>8.83 ***</td>
</tr>
<tr>
<td>Metabolism</td>
<td>0.99</td>
<td>-10.10</td>
<td>0.76</td>
</tr>
<tr>
<td>Constant</td>
<td>131.73 ***</td>
<td>502.28 ***</td>
<td>253.43 ***</td>
</tr>
<tr>
<td>Observations</td>
<td>10,000</td>
<td>10,000</td>
<td>20,000</td>
</tr>
<tr>
<td>Pseudo R²</td>
<td>0.03</td>
<td>0.04</td>
<td>0.18</td>
</tr>
</tbody>
</table>

(*** p < 0.01; ** p < 0.05; * p < 0.10)

Dependent variable: The number of iterations the system endured (Proposition 1)

* I pooled the data (both the security-only world and the security-and-trade world), estimating the fully interacted model and then tested the second group coefficients against zero. The fully interacted model has a dummy variable indicating the group and the extra interaction terms with this dummy variable and other major variables. Thus, the original equation for each separate group and the equation for the pooled dataset (for test) are:

\[
\text{Duration of runs} = \beta_0 + \beta_1 \cdot \text{pow}_sd + \beta_2 \cdot \text{error}_sd + \beta_3 \cdot \text{warcost} \\
+ \beta_4 \cdot \text{wcdispar} + \beta_5 \cdot \text{repar} + \beta_6 \cdot \text{meta}_sd \quad \text{(Original)}
\]

\[
\text{Duration of runs} = \beta_0 + \beta_1 \cdot \text{pow}_sd + \beta_2 \cdot \text{error}_sd + \beta_3 \cdot \text{warcost} \\
+ \beta_4 \cdot \text{wcdispar} + \beta_5 \cdot \text{repar} + \beta_6 \cdot \text{meta}_sd + \beta_7 \cdot g \\
+ \beta_8 \cdot g \cdot \text{pow}_sd + \beta_9 \cdot g \cdot \text{error}_sd + \beta_{10} \cdot g \cdot \text{warcost} \\
+ \beta_{11} \cdot g \cdot \text{wcdispar} + \beta_{12} \cdot g \cdot \text{repar} + \beta_{13} \cdot g \cdot \text{meta}_sd \quad \text{(Pooled)}
\]

where \(g\) is a dummy variable to identify each world (0 = the security-only world and 1 = the security-and-trade world). After running a tobit regression with this pooled data, each interaction term (with \(g\)) is tested.
4.3.2 Balance-of-Power

The cases for the analysis of balance-of-power patterns are collected from the sample runs of 1,000 simulations in the two worlds. In the security-only world, 129,980 cases (in which wars were intended) were generated and about 89 percent of them ended with real wars. On the other hand, in the security-and-trade world, the total number of iterations was 160,531 which included both war and trade. Table 4.6 shows the distribution of the cases of alliance formation. As explained in the research design section, stage 1 and 2 represent the unsuccessful efforts of balance-of-power so that wars (or trades) happen without the formation of alliances. Stage 3 and 4 represent the cases of deterrent balance-of-power in which wars were intended but given up by the initiator due to the formation of a defensive alliance. Stage 1 and 2 were included in the classification of deterrent balance-of-power in Cusack and Stoll's original study, but they were not in this study. Thus we may expect a lower ratio of deterrent balance-of-power in the simulation result of this thesis. Stages 5 through 8 represent wars, part of which can be classified as defensive balance-of-power if the power ratio was changed in favor of the defensive side. We can see that most cases are in stages 1 and 2 in both worlds, even though the security-only world has a higher ratio of stage. The introduction of trade seems to increase the ratio of later stages, so that it affects the system to get into more complicated alliance-forming processes (stages 3 to 8) that might have consequent influences on the patterns of system endurance and balance-of-power.
### Table 4.6: Distribution of the Cases of Alliance Formation

<table>
<thead>
<tr>
<th>Cases of Alliance Formation</th>
<th>Security-Only World</th>
<th>Security-and-Trade World</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N</td>
<td>Percent</td>
</tr>
<tr>
<td>1</td>
<td>96,253</td>
<td>74.05</td>
</tr>
<tr>
<td>2</td>
<td>18,751</td>
<td>14.43</td>
</tr>
<tr>
<td>3</td>
<td>8,107</td>
<td>6.24</td>
</tr>
<tr>
<td>4</td>
<td>6,029</td>
<td>4.64</td>
</tr>
<tr>
<td>5</td>
<td>195</td>
<td>0.15</td>
</tr>
<tr>
<td>6</td>
<td>342</td>
<td>0.26</td>
</tr>
<tr>
<td>7</td>
<td>124</td>
<td>0.10</td>
</tr>
<tr>
<td>8</td>
<td>179</td>
<td>0.14</td>
</tr>
<tr>
<td>Total</td>
<td>129,980</td>
<td>100.00</td>
</tr>
</tbody>
</table>

### Table 4.7: Deterrent Balance-of-Power

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Security-Only World</th>
<th>Security-and-Trade World</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No BOP</td>
<td>Deterrent BOP</td>
</tr>
<tr>
<td>1-100</td>
<td>86,385</td>
<td>13,606</td>
</tr>
<tr>
<td>101-200</td>
<td>29,459</td>
<td>530</td>
</tr>
<tr>
<td>201-300</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>301-400</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>401-500</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>501-600</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>601-700</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>701-800</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>801-900</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>901-1,000</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>115,844</td>
<td>14,136</td>
</tr>
</tbody>
</table>

Table 4.7: Deterrent Balance-of-Power
Table 4.7 shows that the average ratio of deterrent balance-of-power slightly increased from 11 percent to 13 percent when the trade option is introduced. Considering that the security-and-trade world includes both options of war and trade, the ratio of 13 percent of deterrent balance-of-power cases is relatively higher than the ratio of 11 percent in the security-only world. In particular, as the distribution of cases is biased toward the range below iteration 200, the difference in the deterrent balance-of-power ratios between two worlds gets bigger. Within iteration 100, deterrent balance-of-power happened in only 14 percent of cases in the security-only world, while it increased to 23 percent in the security-and-trade world. Between iterations 101 and 200, the ratio increased from 2 percent to 10 percent. However, even in the security-and-trade world, the ratio of deterrent balance-of-power rapidly dropped as the iteration number increased. This seems to reflect the increasing asymmetry among state powers, which we may expect to happen with the reduction of the number of state agents in world politics.

Table 4.8 compares the probit analyses of two worlds, in terms of the explanatory variables' impact on deterrent balance-of-power. While both analyses are based on a large number of observations (129,980 and 165,531, respectively), we have to remind ourselves that the first two stages will not be regarded as (deterrent or defensive) “balance-of-power” because no alliances are formed against the initiator's intention of war. These cases are included in this analysis but they are not coded as balance-of-power, so that they will reduce the impacts of the explanatory variables. Actually, more than 80 percent of cases ended with stage 1 and 2 in both worlds (with no alliance).³⁴

³⁴ About 60 percent and 52 percent of cases in each world, respectively, were excluded from the selection of "defensive balance-of-power."
### Table 4.8: Probit Analysis of Deterrent Balance-of-Power

* Like Table 4.5, the test statistics were produced by the same process, except that a probit analysis is conducted in this case. The original equation for each separate group and the equation for the pooled dataset (for test) are:

\[
\text{Deterrent balance-of-power} = \beta_0 + \beta_1 \cdot \text{i\_group} + \beta_2 \cdot \text{pow.sd} + \beta_3 \cdot \text{error.sd} + \beta_4 \cdot \text{warcost} + \beta_5 \cdot \text{repar} + \beta_6 \cdot \text{meta.sd}
\]

\[
\text{(Original)}
\]

\[
\text{Deterrent balance-of-power} = \beta_0 + \beta_1 \cdot \text{i\_group} + \beta_2 \cdot \text{pow.sd} + \beta_3 \cdot \text{error.sd} + \beta_7 \cdot g \cdot \text{i\_group} + \beta_8 \cdot g \cdot \text{pow.sd} + \beta_9 \cdot g \cdot \text{error.sd} + \beta_{10} \cdot g \cdot \text{warcost} + \beta_{11} \cdot g \cdot \text{repar} + \beta_{12} \cdot g \cdot \text{meta.sd}
\]

\[
\text{(Pooled)}
\]

where \(g\) is a dummy variable to identify each world (0 = the security-only world and 1 = the security-and-trade world). After running a probit analysis with this pooled data, each interaction term (with \(g\)) is tested.
Considering this point, we know that all the control variables – iteration group, power distribution, power estimation error, war cost maximum, reparations, and even metabolism distribution – are significant in their impacts on deterrent balance-of-power. In case of power distribution, the direction of influence changed between the two worlds, while other factors show only the change in the magnitude of the impact. The role of iteration group in the explanatory variable group seems to be reduced while that of power estimation error increases with the introduction of trade. The last column of Table 4.8 shows the result of test for the change in the effects of each variable after the introduction of trade. All variables were significantly changed by the introduction of trade option at $p = 0.01$, except the metabolism rate. Thus we can conclude that the impacts of major control variables on the deterrent balance-of-power are significantly changed in the security-and-trade world.

The analysis of defensive balance-of-power takes a different form – an OLS regression on the change in the power ratio of initiator-to-target before and after the formation of alliances. For this, Cusack and Stoll selected the cases in which real wars occurred. Among all the cases that led to wars, the cases in which the power ratio changes in favor of the defensive side were coded as “defensive balance-of-power.” This study takes the same format in analyzing defensive balance-of-power behaviors. Table 4.9 shows the overall distribution of defensive balance-of-power when a war occurs. Note that the number of cases is reduced both in the security-only world and the security-and-trade world ($N = 840$ and $N = 1,515$, respectively). Among them, only the cases that show a positive change in favor of the target (or the defensive alliance) are coded as “defensive balance-of-power.” We see the overall probabilities of defensive balance-of-
power are 0.40 and 0.37 for the security-only world and the security-and-trade world, respectively. Now, let’s discuss about whether the introduction of trade has any change in the effects of major control variables on the defensive balance-of-power.

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Security-Only World</th>
<th>Security-and-Trade World</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No BOP</td>
<td>BOP</td>
</tr>
<tr>
<td>1-100</td>
<td>478</td>
<td>317</td>
</tr>
<tr>
<td>101-200</td>
<td>28</td>
<td>17</td>
</tr>
<tr>
<td>201-300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>301-400</td>
<td></td>
<td></td>
</tr>
<tr>
<td>401-500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>501-600</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601-700</td>
<td></td>
<td></td>
</tr>
<tr>
<td>701-800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>801-900</td>
<td></td>
<td></td>
</tr>
<tr>
<td>901-1,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>506</td>
<td>334</td>
</tr>
</tbody>
</table>

Table 4.9: Defensive Balance-of-Power

Table 4.10 is the result of OLS regressions on the effect of the set of explanatory variables on the level of difference in the power ratio changes between the initiator/target and the offensive/defensive alliances. As the dependent variable is measured on a continuous scale, OLS regression was used in estimating the effects by the explanatory variables. In both worlds, the initial power ratio, iteration group, and war cost maximum show similar influences on defensive balance-of-power, while power estimation error and
reparations become significant in the security-and-trade world. A bigger difference in initial power distribution seems to cause the frequency of defensive balance-of-power to decrease more in the security-and-trade world. On the other hand, the impact of iteration group seems a little bit decreased in the security-and-trade world. But, from this result, we know that the efforts of defensive balance-of-power are stronger in the later periods than in the earlier stages of the simulation.

In Table 4.10, power estimation error is negatively related to defensive balance-of-power in the security-and-trade world, while the reparations ratio has a positive effect. These two variables show the change in direction in their impacts on the dependent variable. Power distribution, however, does not have any significant influence on the change of power ratio while it has much on the deterrent balance-of-power. Although all the control variables are significant in the probit analysis of deterrent balance-of-power, the factors of power distribution and metabolism are not so in the OLS analysis of defensive balance-of-power. The last column in Table 4.10 illustrates the test result of the change in the role of major control factors. Only power estimation error showed a significant change in its effects on the defensive balance-of-power with the introduction of trade option, at $p = 0.001$. Reparations and initial power ratio also showed significant change in their influence at $p = 0.1$ and $p = 0.05$, respectively. All other variables do not change significantly in their effects on the defensive balance-of-power even at $p = 0.1$ with the trade option. Thus we can conclude that, unlike the case of deterrent balance-of-power, the impacts of major control variables are not much different between the security-only world and the security-and-trade world.
<table>
<thead>
<tr>
<th>Variables</th>
<th>Security-Only World</th>
<th>Security-and Trade World</th>
<th>H₀: Extra Impact = 0.0 F(1,2337)*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Power Ratio</td>
<td>-0.76 ***</td>
<td>-1.22 ***</td>
<td>3.21 *</td>
</tr>
<tr>
<td>Iteration Group</td>
<td>0.63 ***</td>
<td>0.48 ***</td>
<td>0.56</td>
</tr>
<tr>
<td>Power</td>
<td>-0.09</td>
<td>-0.16</td>
<td>0.03</td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>0.73</td>
<td>-4.93 ***</td>
<td>32.88 ***</td>
</tr>
<tr>
<td>War Cost Maximum</td>
<td>2.27 **</td>
<td>2.98 ***</td>
<td>0.20</td>
</tr>
<tr>
<td>Reparations</td>
<td>-0.49</td>
<td>1.63 **</td>
<td>4.79 **</td>
</tr>
<tr>
<td>Metabolism</td>
<td>0.07</td>
<td>-0.13</td>
<td>0.31</td>
</tr>
<tr>
<td>Constant</td>
<td>-0.05</td>
<td>1.24 ***</td>
<td>0.01</td>
</tr>
<tr>
<td>Observations</td>
<td>840</td>
<td>1,515</td>
<td>2,355</td>
</tr>
<tr>
<td>Adjusted R²</td>
<td>0.04</td>
<td>0.15</td>
<td>0.14</td>
</tr>
</tbody>
</table>

(*** p < 0.01; ** p < 0.05; * p < 0.10)

Dependent variable: The dummy coded 1 if defensive balance-of-power happened (a war is intended and happened but the power ratio between the initiator and the target is less favorable to the initiator than the original ratio), and coded 0 otherwise.

**Table 4.10: Regression Analysis of Defensive Balance-of-Power**

* Like Table 4.5 and Table 4.8, the test statistics were produced by the same process, except that an OLS analysis is conducted in this case. The original equation for each separate group and the equation for the pooled dataset (for test) are:

\[
\text{Defensive balance-of-power} = \beta_0 + \beta_1 \cdot \text{it\_group} + \beta_2 \cdot \text{it\_ratio} + \beta_3 \cdot \text{pow\_sd} \\
+ \beta_4 \cdot \text{error\_sd} + \beta_5 \cdot \text{warcost} + \beta_6 \cdot \text{repar} + \beta_7 \cdot \text{meta\_sd} \quad (\text{Original})
\]

\[
\text{Defensive balance-of-power} = \beta_0 + \beta_1 \cdot \text{it\_group} + \beta_2 \cdot \text{it\_ratio} + \beta_3 \cdot \text{pow\_sd} \\
+ \beta_4 \cdot \text{error\_sd} + \beta_5 \cdot \text{warcost} + \beta_6 \cdot \text{repar} + \beta_7 \cdot \text{meta\_sd} \\
+ \beta_8 \cdot g + \beta_9 \cdot g \cdot \text{it\_group} + \beta_{10} \cdot g \cdot \text{it\_ratio} \\
+ \beta_{11} \cdot g \cdot \text{pow\_sd} + \beta_{12} \cdot g \cdot \text{error\_sd} + \beta_{13} \cdot g \cdot \text{warcost} \\
+ \beta_{14} \cdot g \cdot \text{repar} + \beta_{15} \cdot g \cdot \text{meta\_sd} \quad (\text{Pooled})
\]

where g is a dummy variable to identify each world (0 = the security-only world and 1 = the security-and-trade world). After running an OLS analysis with this pooled data, each interaction term (with g) is tested.
4.3.3 State Survival

Proposition 3 refers to the probable effects of the introduction of trade on the state survival pattern at the systemic level. Perhaps we may expect that states survive longer with the more cooperative interaction option (trade) among themselves in terms of their survival rate. Following Cusack and Stoll's original scheme, this section will analyze the simulation results with three dependent variables – survival to iteration 100, survival to iteration 1,000, and the number of iterations during which the state survived. Also a fourth equation will be formed in order to see the impact of an environmental factor, measured by the combined power of neighbor states. All of these relationships are applied only to the states that are generated at the initial stage of simulation runs; other new states that are generated during the runs – by the collapse of multi-territory states – will not be included in this analysis. As such, the total number of relevant cases for this section will be 98,000 that are selected out of 1,000 random sample runs (98 states × 1,000 runs). For a state, the introduction of trade seems to impact on her survival by extending the period of survival further.

---

35 I increased the number of sample runs in order to generate more relevant cases, even though Cusack and Stoll used a small number of cases totaling 1,960 (98 states × 20 runs).
<table>
<thead>
<tr>
<th>Variables</th>
<th>Security-Only World</th>
<th>Security-and-Trade World</th>
<th>Chi-square*</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>State-Level Variables</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Share of System Power at Start</td>
<td>-0.11***</td>
<td>0.02</td>
<td>42.28***</td>
</tr>
<tr>
<td>Error Rate in Estimating Self</td>
<td>-0.21***</td>
<td>-0.42***</td>
<td>53.62***</td>
</tr>
<tr>
<td>Error Rate in Estimating Others</td>
<td>0.32***</td>
<td>0.48***</td>
<td>30.77***</td>
</tr>
<tr>
<td>Geographic Position</td>
<td>0.01***</td>
<td>0.00***</td>
<td>16.57***</td>
</tr>
<tr>
<td><strong>System-Level Variables</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>1.35***</td>
<td>1.44***</td>
<td>0.65</td>
</tr>
<tr>
<td>Power</td>
<td>0.05</td>
<td>0.42***</td>
<td>71.58***</td>
</tr>
<tr>
<td>Reparations</td>
<td>0.37***</td>
<td>0.58***</td>
<td>3.70</td>
</tr>
<tr>
<td>War Cost Maximum</td>
<td>-4.82***</td>
<td>0.01</td>
<td>1,152.53***</td>
</tr>
<tr>
<td>Metabolism</td>
<td>-0.07**</td>
<td>0.00</td>
<td>2.81</td>
</tr>
<tr>
<td>Constant</td>
<td>-0.53***</td>
<td>-0.30***</td>
<td>239.27***</td>
</tr>
<tr>
<td>Observations</td>
<td>98,000</td>
<td>98,000</td>
<td>196,000</td>
</tr>
<tr>
<td>Pseudo R²</td>
<td>0.03</td>
<td>0.01</td>
<td>0.15</td>
</tr>
</tbody>
</table>

Dependent variable: The dummy coded 1 if a state survived to iteration 100, and coded 0 otherwise. (Proposition 3)

**Table 4.11: Probit Analysis of State Survival to Iteration 100**

* The test statistics were produced after running a probit regression. The original equation for each separate group and the equation for the pooled dataset (for test) are:

\[
\text{Survival to iteration 100} = \beta_0 + \beta_1 \cdot \text{rel_pow} + \beta_2 \cdot \text{err}_\text{own} + \beta_3 \cdot \text{err}_\text{oth} + \beta_4 \cdot \text{geo} + \beta_5 \cdot \text{pow_sd} + \beta_6 \cdot \text{error_sd} + \beta_7 \cdot \text{repar} + \beta_8 \cdot \text{warcost} + \beta_9 \cdot \text{meta_sd} \quad \text{(Original)}
\]

\[
\text{Survival to iteration 100} = \beta_0 + \beta_1 \cdot \text{rel_pow} + \beta_2 \cdot \text{err}_\text{own} + \beta_3 \cdot \text{err}_\text{oth} + \beta_4 \cdot \text{geo} + \beta_5 \cdot \text{pow_sd} + \beta_6 \cdot \text{error_sd} + \beta_7 \cdot \text{repar} + \beta_8 \cdot \text{warcost} + \beta_9 \cdot \text{meta_sd} + \beta_{10} \cdot g \cdot \text{rel_pow} + \beta_{11} \cdot g \cdot \text{geo} + \beta_{12} \cdot g \cdot \text{error_sd} + \beta_{13} \cdot g \cdot \text{repar} + \beta_{14} \cdot g \cdot \text{meta_sd} \quad \text{(Pooled)}
\]

where \( g \) is a dummy variable to identify each world (0 = the security-only world and 1 = the security-and-trade world). After running a probit analysis with this pooled data, each interaction term (with \( g \)) is tested.
Table 4.12: Probit Analysis of State Survival to Iteration 1,000

<table>
<thead>
<tr>
<th>Variables</th>
<th>Security-Only World</th>
<th>Security-and-Trade World</th>
<th>Chi-square*</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>State-Level Variables</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Share of System Power at Start</td>
<td>0.06</td>
<td>0.05***</td>
<td>0.08</td>
</tr>
<tr>
<td>Error Rate in Estimating Self</td>
<td>0.58***</td>
<td>0.20***</td>
<td>34.10***</td>
</tr>
<tr>
<td>Error Rate in Estimating Others</td>
<td>-0.61***</td>
<td>0.13***</td>
<td>131.00***</td>
</tr>
<tr>
<td>Geographic Position</td>
<td>0.00</td>
<td>0.01***</td>
<td>27.64***</td>
</tr>
<tr>
<td><strong>System-Level Variables</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>-1.90***</td>
<td>0.95***</td>
<td>131.38***</td>
</tr>
<tr>
<td>Power</td>
<td>-0.63***</td>
<td>0.27***</td>
<td>84.28***</td>
</tr>
<tr>
<td>Reparations</td>
<td>-0.06</td>
<td>0.93***</td>
<td>17.23***</td>
</tr>
<tr>
<td>War Cost Maximum</td>
<td>2.29***</td>
<td>3.64***</td>
<td>18.41***</td>
</tr>
<tr>
<td>Metabolism</td>
<td>-0.43***</td>
<td>0.16***</td>
<td>37.01***</td>
</tr>
<tr>
<td>Constant</td>
<td>-1.88***</td>
<td>-3.01***</td>
<td>492.30***</td>
</tr>
<tr>
<td>Observations</td>
<td>98,000</td>
<td>98,000</td>
<td>196,000</td>
</tr>
<tr>
<td>Pseudo R²</td>
<td>0.04</td>
<td>0.02</td>
<td>0.05</td>
</tr>
</tbody>
</table>

(*** p < 0.01; ** p < 0.05; * p < 0.10)

Dependent variable: The dummy coded 1 if a state survived to iteration 1,000, and coded 0 otherwise. (Proposition 3)

* The original equation for each separate group and the equation for the pooled dataset (for test) are:

Survival to iteration 1,000 = β₀ + β₁ · rel_pow + β₂ · err_own + β₃ · err_oth + β₄ · geo + β₅ · pow_sd + β₆ · error_sd + β₇ · repar + β₈ · warcost + β₉ · meta_sd  

Survival to iteration 1,000 = β₀ + β₁ · rel_pow + β₂ · err_own + β₃ · err_oth + β₄ · geo + β₅ · pow_sd + β₆ · error_sd + β₇ · repar + β₈ · warcost + β₉ · meta_sd + β₁₀ · g · rel_pow + β₁₁ · g · err_own + β₁₂ · g · err_oth + β₁₃ · g · geo + β₁₄ · g · pow_sd + β₁₅ · g · error_sd + β₁₆ · g · repar + β₁₇ · g · warcost + β₁₈ · g · meta_sd

where g is a dummy variable to identify each world (0 = the security-only world and 1 = the security-and-trade world). After running a probit analysis with this pooled data, each interaction term (with g) is tested.
Tables 4.11 through 4.14 are the simulation results of the two worlds regarding the state survival rate. The first three tables divide the explanatory variables into two groups – the state-level variables and the system-level variables. Table 4.11 is the comparison for the state survival rate to a short period of time (iteration 100). We can observe here that the introduction of trade makes every state-level variable change its own impact level on individual state's survival. One noticeable pattern is the reduction of the role by the relative power ratio at the initial stage of run (from −0.11 to 0.02). On the other hand, error-related variables tend to have more impacts in the security-and-trade world, in a negative direction for the case of the error for one's own estimation and in a positive direction for the case of the error for the others. While the role of geographic position seems to change its role not much (from 0.01 to 0.00), but it still is statistically significant. On the other hand, the system-level variables do not show the same changes in their influence – power distribution and war cost maximum changed significantly, reparations and metabolism changed slightly, and power estimation error does not change much with the introduction of the trade option.

Table 4.12 is the result for the 1,000-iteration case. Here, we can see that all variables changed their influence level significantly, except the share of system power at the initial stage. The role of error-related variables has been reduced with the introduction of trade, while that of the geographic position slightly increased. All of the system-level variables significantly changed their impact on state survival ratio with the trade option. This makes us conclude that the system-level variables, such as power estimation error, power distribution, reparations, war cost maximum, and metabolism, tend to have more impact on the state survival ratio in the long term rather than in the short term.
<table>
<thead>
<tr>
<th>Variables</th>
<th>Security-Only World</th>
<th>Security-and-Trade World</th>
<th>F(1,195981)*</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>State-Level Variables</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Share of System Power at Start</td>
<td>-2.51***</td>
<td>6.46***</td>
<td>10.15***</td>
</tr>
<tr>
<td>Error Rate in Estimating Self</td>
<td>3.94**</td>
<td>-42.81***</td>
<td>143.27***</td>
</tr>
<tr>
<td>Error Rate in Estimating Others</td>
<td>-1.61</td>
<td>68.54***</td>
<td>313.18***</td>
</tr>
<tr>
<td>Geographic Position</td>
<td>0.20***</td>
<td>1.51***</td>
<td>158.70***</td>
</tr>
<tr>
<td><strong>System-Level Variables</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>-23.76***</td>
<td>458.55***</td>
<td>971.18***</td>
</tr>
<tr>
<td>Power</td>
<td>-17.84***</td>
<td>133.33***</td>
<td>614.25***</td>
</tr>
<tr>
<td>Reparations</td>
<td>11.01*</td>
<td>118.54***</td>
<td>50.94***</td>
</tr>
<tr>
<td>War Cost Maximum</td>
<td>-44.54***</td>
<td>134.60***</td>
<td>73.33***</td>
</tr>
<tr>
<td>Metabolism</td>
<td>-14.47***</td>
<td>17.43***</td>
<td>27.40***</td>
</tr>
<tr>
<td>Constant</td>
<td>96.64***</td>
<td>26.88***</td>
<td>452.91***</td>
</tr>
<tr>
<td>Observations</td>
<td>98,000</td>
<td>98,000</td>
<td>196,000</td>
</tr>
<tr>
<td>Pseudo R²</td>
<td>0.0001</td>
<td>0.0019</td>
<td>0.05</td>
</tr>
</tbody>
</table>

(*** p < 0.01; ** p < 0.05; * p < 0.10)

Dependent variable: The number of iterations a state survived. (Proposition 3)

Table 4.13: Tobit Analysis of the Number of Iterations Survived

* The original equation for each separate group and the equation for the pooled dataset (for test) are:

\[
\text{Number of iterations survived} = \beta_0 + \beta_1 \cdot \text{rel_pow} + \beta_2 \cdot \text{err_own} \\
+ \beta_3 \cdot \text{err_oth} + \beta_4 \cdot \text{geo} + \beta_5 \cdot \text{pow_sd} + \beta_6 \cdot \text{error_sd} \\
+ \beta_7 \cdot \text{repar} + \beta_8 \cdot \text{warcost} + \beta_9 \cdot \text{meta_sd}\] (Original)

\[
\text{Number of iterations survived} = \beta_0 + \beta_1 \cdot \text{rel_pow} + \beta_2 \cdot \text{err_oth} \\
+ \beta_3 \cdot \text{err_oth} + \beta_4 \cdot \text{geo} + \beta_5 \cdot \text{pow_sd} + \beta_6 \cdot \text{error_sd} \\
+ \beta_7 \cdot \text{repar} + \beta_8 \cdot \text{warcost} + \beta_9 \cdot \text{meta_sd} + \beta_{10} \cdot g \\
+ \beta_{11} \cdot g \cdot \text{rel_pow} + \beta_{12} \cdot g \cdot \text{err_oth} + \beta_{13} \cdot g \cdot \text{warcost} + \beta_{14} \cdot g \cdot \text{meta_sd}\] (Pooled)

where g is a dummy variable to identify each world (0 = the security-only world and 1 = the security-and-trade world). After running a tobit analysis with this pooled data, each interaction term (with g) is tested.

138

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>State-Level Variables</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Share of System Power at Start</td>
<td>0.08*</td>
<td>0.07***</td>
<td>0.04</td>
<td>0.07</td>
<td>0.07***</td>
<td>0.00</td>
</tr>
<tr>
<td>Error Rate in Estimating Self</td>
<td>0.78***</td>
<td>0.06**</td>
<td>106.54***</td>
<td>0.78***</td>
<td>0.08***</td>
<td>80.14***</td>
</tr>
<tr>
<td>Error Rate in Estimating Others</td>
<td>-1.10***</td>
<td>0.11***</td>
<td>277.61***</td>
<td>-1.06***</td>
<td>0.06**</td>
<td>201.04***</td>
</tr>
<tr>
<td>Geographic Position</td>
<td>-0.01***</td>
<td>0.01***</td>
<td>41.64***</td>
<td>-0.01***</td>
<td>0.00***</td>
<td>68.28***</td>
</tr>
<tr>
<td><strong>Environmental Variable</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combined Power of Neighbors</td>
<td></td>
<td></td>
<td></td>
<td>-2.09***</td>
<td>-1.61***</td>
<td>24.18***</td>
</tr>
<tr>
<td>Constant</td>
<td>-1.58***</td>
<td>0.27***</td>
<td>743.57***</td>
<td>-0.63***</td>
<td>0.48***</td>
<td>83.58***</td>
</tr>
<tr>
<td>Observations</td>
<td>20,760</td>
<td>61,206</td>
<td>81,966</td>
<td>20,760</td>
<td>61,206</td>
<td>81,966</td>
</tr>
<tr>
<td>Pseudo R^2</td>
<td>0.05</td>
<td>0.0015</td>
<td>0.24</td>
<td>0.23</td>
<td>0.0069</td>
<td>0.26</td>
</tr>
</tbody>
</table>

(*** p < 0.01; ** p < 0.05; * p < 0.10)

Dependent variable: The dummy coded 1 if a state survived to iteration 200, and coded 0 otherwise. (Proposition 3)

**Table 4.14: Probit Analysis of the Environmental Context for State Survival**

* The original equations for each separate group and the equations for the pooled dataset (for test) are:

- **Survival to iteration 200 = \( \beta_0 + \beta_1 \cdot \text{rel pow} + \beta_2 \cdot \text{err own} \) + \( \beta_3 \cdot \text{err oth} + \beta_4 \cdot \text{geo} \) (Original)
- **Survival to iteration 200 = \( \beta_0 + \beta_1 \cdot \text{rel pow} + \beta_2 \cdot \text{err own} + \beta_3 \cdot \text{geo} \) + \( \beta_4 \cdot \text{err oth} + \beta_5 \cdot \text{geo} \) (Pooled)**
- **Survival to iteration 200 = \( \beta_0 + \beta_1 \cdot \text{rel pow} + \beta_2 \cdot \text{err own} \) + \( \beta_3 \cdot \text{geo} + \beta_4 \cdot \text{neighpow} \) (Original)
- **Survival to iteration 200 = \( \beta_0 + \beta_1 \cdot \text{rel pow} + \beta_2 \cdot \text{err own} + \beta_3 \cdot \text{geo} + \beta_4 \cdot \text{neighpow} \) + \( \beta_5 \cdot \text{err oth} + \beta_6 \cdot \text{geo} + \beta_7 \cdot \text{neighpow} \) (Pooled)**

where \( g \) is a dummy variable to identify each world (0 = the security-only world and 1 = the security-and-trade world). After running probit analyses with these pooled data, each interaction term (with \( g \)) is tested.
Table 4.13 is the result of the tobit analysis on the impact of major explanatory variables on the number of iterations a state survived. Here again, all the variables – both the state-level variables and the system-level variables – show significant changes in their impact on the state survival ratio. Considering that the security-only world ended with the emergence of an empire before the iteration 300, this result should not be a surprise. Table 4.14 is another replication of Cusack and Stoll’s investigation of the environmental variable, measured by the combined power of neighbor countries, with the trade option. In both cases without and with the environmental variable, the trade option significantly decreases the impact of major variables on the state survival ratio, except the initial power ratio.

4.4 Summary

From Figure 4.3 and Table 4.5, we know that the introduction of trade to the security-only world makes the system of states endure longer before collapsing into universal empires. But how much and in what way does it so? This chapter answers this question by articulating and testing Propositions 1, 2, and 3. For the analysis of system endurance (Proposition 1), we observed that the security-only world ended with the emergence of an empire in all runs before iteration 200. This is naturally expected due to fixing the LV sigma value to 1.0, which does not allow any chance factor in determining the winning of war.

Because I could not compare the security-only world with the security-and-trade world directly in terms of system multiplicity due to the non-variance of the binary
dependent variable in the first world, a probit analysis seems to be less reliable. But the side-by-side comparison of different settings (iteration limits) shows that the impact of trade is bigger in some variables — power estimation error, war cost maximum, and disproportionate war cost — while smaller in other variables — power distribution and reparations. System endurance measured by the number of iterations (as far as the system keeps its multiplicity) shows a much heavier impact of trade option than in the previous case of binary dependent variable. Trade makes the system multiplicity appear in a longer term, even though it does not guarantee it to continue forever. The significance test shows that all the major control variables significantly increased their impact on the system multiplicity, except the metabolism rate. Therefore, we can verify Proposition 1 about the impact of the trade option on the system multiplicity and endurance.

The impact of the trade option on the pattern of balance-of-power has mixed effects. Balance-of-power is measured by two indices — deterrent balance-of-power and defensive balance-of-power. One apparent pattern in the case of deterrent balance-of-power is that balancing behavior decreases as time passes, so that it confirms the difficulty of balance-of-power as the number of state agents in the system diminishes. However, it seems that the introduction of trade increases the probability of deterrent balance-of-power. Although the average ratios of deterrent balance-of-power are not much different from each other (0.11 and 0.13, respectively), there are bigger differences between them in the earlier stages of iterations in which most balancing behaviors happen. This point should not be underestimated if we consider the relatively smaller number of war cases in the security-and-trade world. Also, the introduction of trade changes the size and direction of some variables' influence on the occurrence of deterrent balance-of-power.
power. Particularly, the roles of power distribution and power estimation error are important in the increase of the deterrent balance-of-power ratio in the security-and-trade world, although all the major variables change their impact levels with the trade option except the metabolism rate.

In the regression analysis of defensive balance-of-power, the result shows no conspicuous differences between the two worlds, except the higher impact of power estimation error in the security-and-trade world. Overall, the ratios of defensive balance-of-power seem to be not much different between the two worlds (0.40 and 0.37, respectively). The significance test of each major variable shows a similar result—only power estimation error was significantly changed its role at $p = 0.01$. All other variables did not show any significant changes at this probability level even with the introduction of the trade option. Thus we cannot verify Proposition 2 in the case of defensive balance-of-power, even though we can so in the case of deterrent balance-of-power.

The introduction of trade makes the state survival rate increase, especially at the initial stage of runs. The probit analysis of the 100-iteration cases shows that the impact of the trade option is significant more at the state-level variables, such as the initial power ratio, error rate in estimating self and other's power, and geographic position. At the system-level only power distribution and war cost maximum are significant at $p = 0.01$. On the other hand, the same analysis of the 1,000-iteration cases shows that the system-level variables have more significant changes in their role with the introduction of the trade option. On the other hand, the initial power ratio did not change its impact on the state survival ratio even in the security-and-trade world. The results of these two comparisons imply that the system-level factors have more impact on the state survival
ratio in the longer term, while the state-level properties do so in the short term. The tobit analysis of the number of iterations a state survived shows that all the state-level and the system-level variables have bigger impacts on the state survival rate with the introduction of the trade option. The same pattern emerges in the analysis of the impact by the environmental factor (except the initial power ratio), so that we can verify Proposition 3 that the state survival ratio tends to be higher in the security-and-trade world than in the security-only world.
CHAPTER 5

INTERDEPENDENCE, HETEROGENEITY AND WAR

5.1 Research Themes

The dynamics of world politics, especially the war patterns, seem to be influenced by the dyadic factors, too. This chapter analyzes the impact of two dyadic factors – economic interdependence and the heterogeneity of state size (measured by state power) – on the probability of war. Perhaps we might expect that a pair of states with higher economic interdependence level fight less frequently than a dyad of states with lower interdependence. The property of state dyads in terms of the difference in states’ power is also expected to have an impact on the dynamics of war; we anticipate different patterns of war between a dyad of great powers from a dyad of small powers. Also the dyad composed of heterogeneous states (in terms of state power) may have a different pattern of war from the dyad composed of homogeneous states. The ratio of state powers in a dyad can be regarded as another important dyadic property in determining the probability of war, too. As such, these factors – economic interdependence and heterogeneity – between two participating states in a dyad will be investigated in this chapter.
As mentioned in the previous chapter, the effect of interdependence on war (and peace) has been one of the most popular topics in international relations. If we can experiment with this theme using the simulation method, it may contribute to the development of this field by exploring the cases that have not existed but that are theoretically plausible. In the experiment, we can generate many cases with different levels of economic interdependence and measure their impacts on the systemic phenomenon, especially on the probability of war. As the simulation model in this study includes the trade module as well as the war module, a rational state agent can be easily assumed as influenced by the level of economic interdependence and the level of power difference in the dyad in her choice. For the factor of interdependence, we can imagine a more peaceful world in terms of war frequency with a higher level of interdependence at the dyad-level. Thus the first theme to test in this chapter is Proposition 4.

Proposition 4: The more economically interdependent states are, the lower the probability of war between them becomes.

Another theme about the heterogeneity of state size in a dyad is investigated in Propositions 5, 6, and 7. Proposition 5 is about the difference between the dyads of great powers and the dyads of small powers. According to existing theories, the behavior of great (or major) powers is different from that of small (or minor) powers. Proposition 6 tests the impact of the homogeneity (or heterogeneity) of states’ power levels in a dyad on the frequency of war. We may expect less or more frequency in wars in the dyad composed of a major state and a minor state than the dyad with relatively homogeneous
states. While Proposition 5 and 6 deal with the discrete property of dyads – whether the state agents are major states or minors, Proposition 7 uses a continuous variable of power ratio to test the same theme. Particularly, it presupposes that the bigger the power difference between states in a dyad is, the higher the war probability between them.

**Proposition 5:** The probability of war between major states is lower than that between minor states.

**Proposition 6:** The probability of war in heterogeneous dyads (in terms of state power) is higher than that in homogeneous dyads.

**Proposition 7:** The bigger the difference in power between two states (on a continuous scale), the higher the probability of war between them.

### 5.2 Research Design

#### 5.2.1 Variables

The variable of economic interdependence in this chapter needs to be elaborated in its operationalization more than others used in the previous chapter. First of all, the concept has not yet been standardized in the academic circle, so that it is not easy to define and measure it in a theoretically meaningful way. However, as discussed before, the accumulation of empirical studies in this field have enabled us to agree on the concept.
of interdependence at least in the economic area. Old studies, such as Polachek (1980) and Polachek and associates (1982) used a simple index of "trade" measured by the sum of import and export. The concept of "economic dependence," as a broader notion than that trade, has been operationalized by Oneal and Russett (1997, 1999a, 1999b, 1999c) by summing import and export divided by the GDP level in a country. In particular, these scholars used the value of the "lower" dependence level in a dyad as the index of "economic interdependence" for their study of democratic peace. They assume that the "less-constrained state has the greater influence on the likelihood of dyadic conflict" and that the "less-dependent state should have greater freedom to initiate conflict because its economic costs would be less and the beneficial influence of trade as communication would be less" (Oneal and Russett, 1997, 276).

In this thesis, the level of economic interdependence is measured by the similar index suggested by Oneal and Russett, but there is a slight difference. Due to the simulation scheme, the economic dependence of each state is measured by the "expected utility of trade" in a dyad. As explained in the chapter of research method, a state calculates its own expected utility of trade and war, separately, toward each of her neighbors. The expected utility of trade is calculated by the open information of the partner's metabolism and level of the two resources as well as her own. This complex process of calculation of expected utility is to be finished for all individual states before any actor is selected as the initiator. Thus, any state holds a level of expected utility of

---

36 Oneal and Russett admit that trade is not a perfect indicator of economic interdependence, but they argue that it is a reasonable index as the trade-to-GDP ratio shows a country's dependence on its trading partners and other indices such as foreign investment follow the same pattern as trade (Oneal and Russett, 1997, 275; especially see footnote 11).

147
trade as well as that of war toward each of her possible interaction partners. While the expected utility of war may be negative in many cases due to the cost of war and probable indemnities after the war, this study assumes that there is no negative expected utility of trade. As mentioned before, trade is assumed to provide “extra” benefits to each partner in a dyad that might not be available otherwise.

As such, it is reasonable to use the level of expected utility of trade for a state in a dyad as the index of economic dependence toward the other. A more dependent state will choose the trade option toward the other in the dyad, even though she might expect positive expected utility of war. Of course, a more powerful state (i.e., a country holding more resources or lower metabolism rates) tends to be less dependent than the smaller partner, so that the expected utility of trade will be weighted by each state’s power level. For this reason, the expected utility of trade for a large country toward a very small country may be close to zero, even though it should never be negative. So the level of economic interdependence in a dyad is to be measured by the comparison of each state’s expected utility of trade weighted by power level. Here, I will use Oneal and Russett’s criterion of “lower dependence level” for the economic interdependence for the same reason they suggested. Therefore, the level of economic interdependence in a dyad will be measured by the “lower value of expected utility of trade divided by power level of each state in the dyad.”

The operationalization of the concepts used in the second theme – heterogeneity – seems more complex as there is less information to use from any theoretical or empirical sources. Here, the meaning of “major” and “minor” should be clarified before we proceed to define the property of dyads. The propositions used in this study consider the
difference in state power as a major factor, so that we may use “strong” or “great” vis-à-vis “weak” or “small” in order to classify states in terms of power. Whatever terms are used, therefore, there is no big difference in their implications as far as their operationalizations are the same. This study chooses to use the terms “major” and “minor” in delineating the power difference in a dyad. If a state is defined as “major” it means that the state is regarded as a great power (or a strong power) in the system-level (not just at the dyad-level). The term “minor” is applied to all other states which cannot be classified as “major.”

The problem is in the criterion of “major” and “minor” because there is no clear criterion in theory and in empirical studies. It may be a very subjective and fluctuating criterion, if any, to tell the difference between major and minor powers in world politics. One of the few resources that used this kind of classification is COW, in which some countries are marked as “major” based upon the “intercoder agreement” (Singer and Small, 1972, 23). Table 5.1 shows the composition of major powers in world politics during the period between 1816 and 1965. We can observe that the total number of major powers in world politics has been between 4 and 8. The composition of major powers seems not to be different from existing studies, even though the authors of COW did not show any precise criterion for their classification. Our common sense also makes us acknowledge the reasonableness in the authors’ and coders’ selection of major powers in COW.

37 According to Singer and Small, there was a “high scholarly consensus” on the composition of the oligarchy of major powers up to World War II, while it was not so after that (1972, 23).
<table>
<thead>
<tr>
<th>Period -</th>
<th>N</th>
<th>USA</th>
<th>UK</th>
<th>France</th>
<th>Germany</th>
<th>Austria-Hungary</th>
<th>Italy</th>
<th>Russia (USSR)</th>
<th>China</th>
<th>Japan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1816</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1859</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1860</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1894</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1895</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1898</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1899</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1917</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1918</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1919</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1921</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1922</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1924</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1925</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1940</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1941</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1943</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1944</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1945</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1946</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1949</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1950</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1965</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Source: COW)

Table 5.1: Composition of Major States, 1816-1965
This thesis will follow the COW dataset in the definition of major power, but another problem is the conversion of the COW’s empirical convention into a formal one. For the simulation study in this thesis, I have to measure the amount of power every major state possesses at every time period. The COW dataset does not clearly describe this quantitative criterion, but it must be available by manual calculation of power shares used. First of all, I calculated the world total of each of the six power indices – military personnel, military expenditure, iron and steel usages, energy usages, total population, and urban population – used in the dataset. Every major power’s relative share of these indices is calculated (for scaling) and summed up across the indices. The summed value is again divided by six (the number of indices) which produces a number that is regarded as the average power index of each state.\textsuperscript{38} Table in Appendix 1 shows the ratio of each major state’s relative power measured in this way (while ignoring non-major powers). In the table, we observe that the smallest major power in a specific year holds roughly 2~3 percent of world total. Thus it is reasonable for us to classify a state as “major” if its relative power takes 3 percent or more of world total. Also, considering that the number of major powers in world politics has been between four and eight, I will set the maximum number of major power group at eight.\textsuperscript{39} This will make a state rejected in the major power group if the number of group already reaches eight, even though her power ratio is over 3 percent. Thus, in my simulation, a state will be entitled “major” if her relative power is over 3 percent and its ranking is within the eighth.

\textsuperscript{38} By this method, the power index out of six raw indices becomes standardized within zero and 1.0. If there is any missing value in the six main indices of power, the total value of state power is divided by six minus the number of missing values.

\textsuperscript{39} This is consistent with Waltz’s argument that eight major states at most have existed since the Treaty of Westphalia, forming an oligarchy of small-number systems (Waltz, 1979, 131).
5.2.2 Equations

There are four propositions to test in this chapter, regarding the influence of dyadic properties – economic interdependence and the heterogeneity of state size – on the probability of war. They are expressed here again:

Proposition 4: The more economically interdependent states are, the lower the probability of war between them becomes.

Proposition 5: The probability of war between major states is lower than that between minor states.

Proposition 6: The probability of war in heterogeneous dyads (in terms of state power) is higher than that in homogeneous dyads.

Proposition 7: The bigger the difference in power between two states (on a continuous scale), the higher the probability of war between them.

The propositions about economic interdependence (Proposition 4) and about the impact of power difference (Proposition 7) need probit equations to test. Proposition 5 and Proposition 6 will be tested with the t-test method in order to see if any difference exists between the two separate groups. Proposition 4 focuses on the impact of economic interdependence on the probability of war, so that the group of explanatory variables for...
the equation will include the level of economic interdependence, in addition to the six main control variables – power distribution, error distribution, war cost maximum, disproportionate war cost, reparations ratio, and metabolism ratio. Also I will formulate another equation with the addition of the iteration group factor in order to see whether there is any difference across different stages of state interactions. Also, Proposition 7 focuses on the impact of the power difference on the probability of war, so that the power ratio between the initiator and the target in the dyad will be included in the group of explanatory variables. As such, the first and the second equations are for Proposition 4 and the last equation is for Proposition 7, while Propositions 5 and 6 do not need equations in these forms:

\[ Pr(War) = \beta_0 + \beta_1 \cdot interdependence + \beta_2 \cdot power_sd + \beta_3 \cdot error_sd + \beta_4 \cdot warcost + \beta_5 \cdot wcdispar + \beta_6 \cdot repar + \beta_7 \cdot meta_sd \]

\[ Pr(War) = \beta_0 + \beta_1 \cdot interdependence + \beta_2 \cdot i\_group + \beta_3 \cdot power_sd + \beta_4 \cdot error_sd + \beta_5 \cdot warcost + \beta_6 \cdot wcdispar + \beta_7 \cdot repar + \beta_8 \cdot meta_sd \]

\[ Pr(War) = \beta_0 + \beta_1 \cdot power\_ratio + \beta_2 \cdot power_sd + \beta_3 \cdot error_sd + \beta_4 \cdot warcost + \beta_5 \cdot wcdispar + \beta_6 \cdot repar + \beta_7 \cdot meta_sd \]

(where \textit{interdependence} is the level of economic interdependence, \textit{i\_group} is the collapsed criteria of time periods from 1 to 1,000, \textit{power\_ratio} is the power ratio between the initiator and the target, \textit{pow\_sd} is the standard deviation of power distribution, \textit{error\_sd} is the standard deviation of power estimation error, \textit{warcost} is the maximum proportion of power to lose in a war, \textit{wcdispar} is the prewar power ratio to asymmetric war cost, \textit{repar} is the proportion of the loser’s power given to winners, and \textit{meta\_sd} is the standard deviation of metabolism distribution)

\[40\] Proposition 5 and Proposition 6 will be tested with t-test method (See Table 5.11 and Table 5.12)
5.3 Analysis of Results

5.3.1 Interdependence and War

Experiments were conducted with 5,000 runs of simulations, which generated a total of 2.8 million individual dyads in which state interactions are initiated or happened in a dyad. For Proposition 4, I selected the cases in which both partners in a dyad have positive expected utilities toward each other. The total number of relevant cases reaches 389,454. Considering that the calculation of expected utility (of trade) is based on the power level of each state, it is not surprising that more than 25 percent of cases are distributed around the economic interdependence level below 0.1. In a few cases the level of interdependence passes over 1.0 (the range of values is between 0.00002 and 3.42) but most are below 0.3. It means that most states (the less constrained partner in a dyad) have expected utilities toward its dyad partner, that are about 30 percent or less vis-à-vis their current power levels. Figure 5.1 shows the distribution of interdependence level.

---

41 If a state in a dyad does not have any positive expected utility (of trade) toward the other, then we cannot regard this dyad as “economically interdependent.” A very small amount of expected trade volume, based upon the calculation of the MRS’s and available resources of both trading partners, can make a state give up trade, according to the trade algorithm used in this study. The following table shows the distribution of cases according to the level of economic interdependence, in which the cases of zero expected utility of trade as “independent” while positive expected utility of trade is marked as “dependent”:

<table>
<thead>
<tr>
<th>Type of Dyad (Initiator – Target)</th>
<th>N</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent – Independent</td>
<td>1,652</td>
<td>0.00058</td>
</tr>
<tr>
<td>Independent – Dependent</td>
<td>18</td>
<td>0.0000063</td>
</tr>
<tr>
<td>Dependent – Independent</td>
<td>2,466,624</td>
<td>0.86</td>
</tr>
<tr>
<td>Dependent – Dependent</td>
<td>389,454</td>
<td>0.14</td>
</tr>
</tbody>
</table>

Distribution of Cases

154

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 5.1: Distribution of Expected Utility

<table>
<thead>
<tr>
<th>Interdependence Level</th>
<th>War Ratio</th>
<th>Trade Ratio</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0 ~ 0.1</td>
<td>18.1</td>
<td>80.8</td>
<td>106,961</td>
</tr>
<tr>
<td>0.1 ~ 0.2</td>
<td>20.5</td>
<td>78.9</td>
<td>90,679</td>
</tr>
<tr>
<td>0.2 ~ 0.3</td>
<td>21.5</td>
<td>78.1</td>
<td>73,705</td>
</tr>
<tr>
<td>0.3 ~ 0.4</td>
<td>11.2</td>
<td>88.5</td>
<td>47,481</td>
</tr>
<tr>
<td>0.4 ~ 0.5</td>
<td>2.3</td>
<td>97.6</td>
<td>27,692</td>
</tr>
<tr>
<td>0.5 ~ 0.6</td>
<td>0.2</td>
<td>99.8</td>
<td>16,292</td>
</tr>
<tr>
<td>0.6 ~ 0.7</td>
<td>0.0</td>
<td>100.0</td>
<td>9,839</td>
</tr>
<tr>
<td>0.7 ~ 0.8</td>
<td>0.0</td>
<td>100.0</td>
<td>6,065</td>
</tr>
<tr>
<td>0.8 ~ 0.9</td>
<td>0.0</td>
<td>100.0</td>
<td>3,715</td>
</tr>
<tr>
<td>0.9 ~ 1.0</td>
<td>0.0</td>
<td>100.0</td>
<td>2,405</td>
</tr>
<tr>
<td>1.0 +</td>
<td>0.0</td>
<td>100.0</td>
<td>4,620</td>
</tr>
<tr>
<td>Total N*</td>
<td>59,782</td>
<td>327,421</td>
<td>389,454</td>
</tr>
</tbody>
</table>

* Including 2,251 cases of no action.

Table 5.2: Interdependence Levels and War/Trade Ratios
The ratio of war across different interdependence level is shown in Table 5.2, along with the ratio of trade. Both ratios do not need to sum up 100 percent because of the failed trials of war and trade. That is, if the initiator finds that the defensive alliance is stronger than her while she cannot find any candidate for an offensive alliance, then she gives up the war effort. In the case of trade, if the initiator finds that the real trade unit is less than a certain value (the minimum is set to 0.00001) then trade does not happen. In the table, we can see that the probability of war is around 20 percent below the interdependence level of 0.3. Interestingly, war happens more frequently with higher interdependence levels up to 3.0 but its frequency decreases after that. The mirror image of trade shows the same trend – the trade ratio drops with higher interdependence level up to 3.0, and then increases after that. This may be because of the size effect – that is, a state can get more benefits from war with the other in the dyad if the interdependence level in the dyad increases within a certain range (below 0.3 here). Thus we can see that the level of peace does not linearly increase with the increase of interdependence between dyadic partners.

Now it’s time for us to run a probit analysis for the impact of the dyadic interdependence level on the war probability. The left two columns of Table 5.3 show the result, including the conventional major control variables – power distribution, error distribution, war cost maximum, disproportionate war cost, reparations, and metabolism ratio. For the cases numbered 389,454, the coefficient of interdependence is −1.89 (without the squared term of interdependence) which is significant enough for us to prove Proposition 4. Other control factors show mixed levels of influence: power distribution, war cost maximum and disproportionate war cost have positive effects on the probability
of war, while the error distribution has negative effects. On the other hand, reparations and the metabolism distribution do not have any significant impacts on the occurrence of war in a dyad. With the squared term, we know that the relationship between the probability of war and the level of interdependence is not linear. This observation matches that of Table 5.2.

It is also interesting to see the impact of the iteration stages on the probability of war occurrence. We may expect that the stage of world politics – in which the dyad is selected – might have impacts on the probability of war. Table 5.4 illustrates the patterns of war and trade across iteration groups. We can see that there is a big difference in the probabilities (of war and trade) in the iteration group, which is a collapsed variable (by the size of 100) of iterations where the dyad is selected. The right two columns of Table 5.3 are the results of probit analyses, including the iteration group variable in the equation without and with the squared term. The impact of interdependence level seems almost the same as that in the model without the iteration group variable (−1.89 and −1.93, respectively). The coefficient of the iteration group is −0.09, which is significant at \( p = 0.01 \), so that we know that the probability of war decreases as iteration processes go on.
<table>
<thead>
<tr>
<th>Variables</th>
<th>Without Iteration Group</th>
<th>With Iteration Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Without I²</td>
<td>With I²</td>
</tr>
<tr>
<td>Interdependence</td>
<td>-1.89 ***</td>
<td>4.42 ***</td>
</tr>
<tr>
<td>Interdependence² (I²)</td>
<td>-14.04 ***</td>
<td>-14.09 ***</td>
</tr>
<tr>
<td>Iteration Group</td>
<td>-0.09 ***</td>
<td>-0.08 ***</td>
</tr>
<tr>
<td>Power</td>
<td>0.50 ***</td>
<td>0.53 ***</td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>-0.86 ***</td>
<td>-0.94 ***</td>
</tr>
<tr>
<td>War Cost Maximum</td>
<td>0.74 ***</td>
<td>0.77 ***</td>
</tr>
<tr>
<td>Disproportionate War Cost</td>
<td>0.52 ***</td>
<td>0.53 ***</td>
</tr>
<tr>
<td>Reparations</td>
<td>0.00</td>
<td>-0.02</td>
</tr>
<tr>
<td>Metabolism</td>
<td>0.00</td>
<td>-0.01</td>
</tr>
<tr>
<td>Constant</td>
<td>-0.84 ***</td>
<td>-1.29 ***</td>
</tr>
<tr>
<td>Observations</td>
<td>389,454</td>
<td>389,454</td>
</tr>
<tr>
<td>Pseudo R²</td>
<td>0.05</td>
<td>0.08</td>
</tr>
</tbody>
</table>

(*** p < 0.01; ** p < 0.05; * p < 0.10)

Table 5.3: Impact of Economic Interdependence

<table>
<thead>
<tr>
<th>Iteration Groups</th>
<th>No Action</th>
<th>Trade Ratio</th>
<th>War Ratio</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than 100</td>
<td>0.9</td>
<td>81.5</td>
<td>17.7</td>
<td>27,305</td>
</tr>
<tr>
<td>101 ~ 200</td>
<td>0.9</td>
<td>80.4</td>
<td>18.7</td>
<td>79,418</td>
</tr>
<tr>
<td>201 ~ 300</td>
<td>0.8</td>
<td>80.0</td>
<td>19.2</td>
<td>83,244</td>
</tr>
<tr>
<td>301 ~ 400</td>
<td>0.5</td>
<td>83.0</td>
<td>16.5</td>
<td>66,404</td>
</tr>
<tr>
<td>401 ~ 500</td>
<td>0.3</td>
<td>87.2</td>
<td>12.6</td>
<td>48,692</td>
</tr>
<tr>
<td>501 ~ 600</td>
<td>0.2</td>
<td>89.7</td>
<td>10.2</td>
<td>33,760</td>
</tr>
<tr>
<td>601 ~ 700</td>
<td>0.2</td>
<td>91.7</td>
<td>8.1</td>
<td>21,640</td>
</tr>
<tr>
<td>701 ~ 800</td>
<td>0.1</td>
<td>92.7</td>
<td>7.2</td>
<td>14,032</td>
</tr>
<tr>
<td>800 ~ 900</td>
<td>0.1</td>
<td>94.0</td>
<td>5.9</td>
<td>9,084</td>
</tr>
<tr>
<td>901 ~ 1,000</td>
<td>0.0</td>
<td>94.8</td>
<td>5.2</td>
<td>5,875</td>
</tr>
<tr>
<td>Total N</td>
<td>2,251</td>
<td>327,421</td>
<td>59,782</td>
<td>389,454</td>
</tr>
</tbody>
</table>

Table 5.4: Iteration Group and War/Trade Ratios

158

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
5.3.2 Heterogeneity and War

For the convenience of analysis, all the dyads generated from sample simulation runs were classified into four types – major to major, major to minor, minor to major, and minor to minor – according to the criterion mentioned in the research design section. Among the total 2.8 million cases of interaction, about 60 percent were initiated by a major state who selected a minor state as a target. The number of dyads between major states is about 16 percent. About 22 percent of cases were between minor states, while the percentage of cases initiated by a minor state toward a major state was slightly less than four. Considering that the initiation process weighs the relative power of each state in the selection of an initiator (with a random number), the high ratio of cases initiated by a major state seems reasonable. Figure 5.2 shows the overall distribution of the cases.

Table 5.5 shows the ratios of war, trade, and no action in each dyad group. In the dyad group composed of two major states, the war ratio is about 23 percent of cases, while it is about 51 percent in dyads of minor states. Major countries tend to be more prudent in their decision to choose war due to the high proportion of expected utility of trade between themselves. Minor states seem to be less induced to trade with each other due to the small amount of probable benefits of trade between them relative to the benefits of war. The dyad initiated by a major state to a minor state has a lower war ratio (about 16 percent) than the dyad with two major states. This implies that the war cost burden of big countries constrains their war ambitions, regardless of target. The trend of minor states’ lowest war ratio (less than 1 percent) against major states is reasonable, with the very low likelihood of winning.
Figure 5.2: Heterogeneity in State Size

<table>
<thead>
<tr>
<th>Heterogeneity</th>
<th>No Action</th>
<th>Trade Ratio</th>
<th>War Ratio</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major to Major</td>
<td>0.2</td>
<td>76.9</td>
<td>22.9</td>
<td>461,444</td>
</tr>
<tr>
<td>Major to Minor</td>
<td>0.1</td>
<td>83.9</td>
<td>16.0</td>
<td>1,651,854</td>
</tr>
<tr>
<td>Minor to Major</td>
<td>0.3</td>
<td>98.8</td>
<td>0.9</td>
<td>106,959</td>
</tr>
<tr>
<td>Minor to Minor</td>
<td>16.5</td>
<td>32.6</td>
<td>50.9</td>
<td>637,221</td>
</tr>
<tr>
<td>N</td>
<td>107,613</td>
<td>2,054,975</td>
<td>694,890</td>
<td>2,587,478</td>
</tr>
</tbody>
</table>

Table 5.5: Heterogeneity and War Ratio
Upon the observation that there is a big difference in the probability of war across different types of dyads, we need to re-check the impact of economic interdependence. Perhaps, we may have different levels of effects by economic interdependence on the probability of war. For testing this sub-level hypothesis, I ran probit analyses on the four sets of dyads separately. Tables 5.6, 5.7, 5.8 and 5.9 show the results of the major-major dyads, the major-minor dyads, the minor-major dyads, and the minor-minor dyads, respectively. The cases (N = 389,454) were used as selected in the previous section, in which both states on a dyad have positive expected utilities toward the other.42

In case of the major-major dyad, the coefficient is bigger than the overall average (-2.27 vs. -1.89). On the other hand, the minor-minor dyad shows less impact by economic interdependence (-1.97). The minor-major dyad (-2.27) shows the same effect as that of the major-major dyad. The lowest level of coefficient in the major-minor dyad (-1.85) seems reasonable when a bigger state initiates action against a smaller state. Thus, at least in the artificial world of this study, major states are more sensitive to the level of economic interdependence among themselves than minor states. Interestingly, major states were more prudent in their initiation of war. One possible reason for this pattern is the high ratio of war cost payments, regardless of the victory. In case of large states, the amount of the war cost is much bigger than the expected gain from war, so that they would not initiate a war unless the target does not hold a certain size of power. Overall, the relationship between the level of interdependence and the probability of war is negative, while we have noticed that the patterns happen different across types of dyad.

42 The total number of cases for the analysis of this section about the dyad heterogeneity is 2.85 million. However, due to the operationalization of interdependence, I chose the cases of interdependence totaling 389,454 for Table 5.6 through Table 5.9. The later part of this section uses the whole dataset again.

161
### Table 5.6: Impact of Economic Interdependence in Major-Major Dyads

<table>
<thead>
<tr>
<th>Variables</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>Economic Interdependence</td>
<td>-2.27***</td>
</tr>
<tr>
<td>Power</td>
<td>0.68 ***</td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>-0.20 **</td>
</tr>
<tr>
<td>War Cost Maximum</td>
<td>0.88 ***</td>
</tr>
<tr>
<td>Disproportionate War Cost</td>
<td>0.65 ***</td>
</tr>
<tr>
<td>Reparations</td>
<td>0.12</td>
</tr>
<tr>
<td>Metabolism</td>
<td>0.00</td>
</tr>
<tr>
<td>Constant</td>
<td>-1.05 ***</td>
</tr>
<tr>
<td>Observations</td>
<td>134,822</td>
</tr>
<tr>
<td>Pseudo $R^2$</td>
<td>0.07</td>
</tr>
</tbody>
</table>

(*** $p < 0.01$; ** $p < 0.05$; * $p < 0.10$)

Dependent variable: The dummy coded 1 if a war happens, and coded 0 otherwise.  
(This is a pre-test for Propositions 5, 6 and 7)

### Table 5.7: Impact of Economic Interdependence in Major-Minor Dyads

<table>
<thead>
<tr>
<th>Variables</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>Economic Interdependence</td>
<td>-1.85 ***</td>
</tr>
<tr>
<td>Power</td>
<td>0.73 ***</td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>-1.22 ***</td>
</tr>
<tr>
<td>War Cost Maximum</td>
<td>1.51 ***</td>
</tr>
<tr>
<td>Disproportionate War Cost</td>
<td>0.51 ***</td>
</tr>
<tr>
<td>Reparations</td>
<td>0.01</td>
</tr>
<tr>
<td>Metabolism</td>
<td>-0.03</td>
</tr>
<tr>
<td>Constant</td>
<td>-0.72 ***</td>
</tr>
<tr>
<td>Observations</td>
<td>162,688</td>
</tr>
<tr>
<td>Pseudo $R^2$</td>
<td>0.05</td>
</tr>
</tbody>
</table>

(*** $p < 0.01$; ** $p < 0.05$; * $p < 0.10$)

Dependent variable: The dummy coded 1 if a war happens, and coded 0 otherwise.  
(This is a pre-test for Propositions 5, 6 and 7)
### Table 5.8: Impact of Economic Interdependence in Minor-Major Dyads

<table>
<thead>
<tr>
<th>Variables</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>Economic Interdependence</td>
<td>-2.27***</td>
</tr>
<tr>
<td>Power</td>
<td>0.10</td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>4.14***</td>
</tr>
<tr>
<td>War Cost Maximum</td>
<td>-1.67***</td>
</tr>
<tr>
<td>Disproportionate War Cost</td>
<td>0.16</td>
</tr>
<tr>
<td>Reparations</td>
<td>-0.42</td>
</tr>
<tr>
<td>Metabolism</td>
<td>-0.08</td>
</tr>
<tr>
<td>Constant</td>
<td>-2.72***</td>
</tr>
<tr>
<td>Observations</td>
<td>47,340</td>
</tr>
<tr>
<td>Pseudo R²</td>
<td>0.07</td>
</tr>
</tbody>
</table>

Dependent variable: The dummy coded 1 if a war happens, and coded 0 otherwise.
(This is a pre-test for Propositions 5, 6 and 7)

### Table 5.9: Impact of Economic Interdependence in Minor-Minor Dyads

<table>
<thead>
<tr>
<th>Variables</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>Economic Interdependence</td>
<td>-1.97***</td>
</tr>
<tr>
<td>Power</td>
<td>-0.21***</td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>0.10***</td>
</tr>
<tr>
<td>War Cost Maximum</td>
<td>-1.13***</td>
</tr>
<tr>
<td>Disproportionate War Cost</td>
<td>0.39***</td>
</tr>
<tr>
<td>Reparations</td>
<td>-0.03</td>
</tr>
<tr>
<td>Metabolism</td>
<td>0.07</td>
</tr>
<tr>
<td>Constant</td>
<td>0.95***</td>
</tr>
<tr>
<td>Observations</td>
<td>44,604</td>
</tr>
<tr>
<td>Pseudo R²</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Dependent variable: The dummy coded 1 if a war happens, and coded 0 otherwise.
(This is a pre-test for Propositions 5, 6 and 7)
Proposition 5 describes the relationship between the likelihood of war and the different types of dyads – the dyads of major states and the dyads of minor states. A total of about 1 million samples were drawn for these two classes of dyads and about 42 percent of them are major-major dyads and the remaining 58 percent are minor-minor dyads. Before the t-test, a Levene test for the same variances for the two group was conducted: its result shows that the variances of the two groups are not the same. Then I ran a t-test with unequal variances. In the result of Table 5.10, the null hypothesis of the same average in the probability of war between the two groups is rejected. So we can say that the probability of war in the minor power dyads is significantly higher than that of the major power dyads.

<table>
<thead>
<tr>
<th>Group</th>
<th>Observations</th>
<th>Mean</th>
<th>Standard Error</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major Power Dyads</td>
<td>461,444</td>
<td>0.2288</td>
<td>0.0006</td>
<td>0.4201</td>
</tr>
<tr>
<td>Minor Power Dyads</td>
<td>637,221</td>
<td>0.5090</td>
<td>0.0006</td>
<td>0.4999</td>
</tr>
<tr>
<td>Combined</td>
<td>1,098,665</td>
<td>0.3913</td>
<td>0.0005</td>
<td>0.4881</td>
</tr>
<tr>
<td>Difference</td>
<td>-0.2802</td>
<td></td>
<td>0.0009</td>
<td></td>
</tr>
</tbody>
</table>

**Levene test for the same variances**

The Levene statistic (F): 264,393.2
Degrees of freedom: 1,098,663
Pr > F = 0

**Two-Sample t-test with unequal variances**

Degrees of freedom: 1,098,663
Diff = Mean of major power dyads - Mean of minor power dyads
H₀: Diff = 0, H₁: Diff ≠ 0
t = -318.3953, Pr > |t| = 0.0000

**Table 5.10: Wars in Major Power Dyads and Minor Power Dyads**

164
Another t-test result for Proposition 6 is described in Table 5.11. Here, the test focuses on the relationship between the homogeneous dyads and the heterogeneous dyads. For this test, I divided the whole cases of 2.8 million into these two groups. The major-major and the minor-minor dyads are classified as the homogeneous group, and the major-minor and the minor-major dyads as the heterogeneous dyads. The hypothesis is that the heterogeneous dyads have a higher probability of war than the homogeneous dyads. A Levene test shows that the variances of the two groups are not the same, so that I ran a t-test for the same average with unequal variance option. The test result verifies that the mean probabilities of the two groups are not the same. Thus we can conclude that the probability of war in the homogeneous dyads (0.39) is significantly higher than that in the heterogeneous dyads (0.15). So we reject Proposition 6.

<table>
<thead>
<tr>
<th>Group</th>
<th>Observations</th>
<th>Mean</th>
<th>Standard Error</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heterogeneous Dyads</td>
<td>1,758,813</td>
<td>0.1506</td>
<td>0.0003</td>
<td>0.3577</td>
</tr>
<tr>
<td>Homogeneous Dyads</td>
<td>1,098,665</td>
<td>0.3913</td>
<td>0.0005</td>
<td>0.4881</td>
</tr>
<tr>
<td>Combined</td>
<td>2,857,478</td>
<td>0.2432</td>
<td>0.0003</td>
<td>0.4290</td>
</tr>
<tr>
<td>Difference</td>
<td></td>
<td>-0.2407</td>
<td>0.0005</td>
<td></td>
</tr>
</tbody>
</table>

**Levene test for the same variances**

The Levene statistic (F): 768,754.52  
Degrees of freedom: 2,857,476  
Pr > F = 0

**Two-sample t-test with unequal variances**

Degrees of freedom: 2,857,476  
\( \text{Diff} = \text{Mean of major power dyads} - \text{Mean of minor power dyads} \)  
\( H_0: \text{Diff} = 0, \quad H_1: \text{Diff} \neq 0 \)  
\( t = -447.3542, \quad \text{Pr} > |t| = 0.0000 \)

**Table 5.11: Wars in Heterogeneous Dyads and Homogeneous Dyads**

165
Proposition 7 states the expected relationship between the power ratio and the probability of war, so that it extends the previous two propositions with a continuous variable. The power ratio measures the power of the initiator divided by the power of the target. The left column of Table 5.12 shows the result of a probit analysis without and with iteration control. For the 2.85 million cases, the overall impact of the power ratio is negative in the frequency of war in the dyad. The result of the table makes us reject Proposition 7 – about the positive relationship between the power ratio (the initiator’s power to the target’s power) and the frequency of war. The same pattern happens whether we include the iteration group variable or not.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Without Iteration Group</th>
<th>With Iteration Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Ratio</td>
<td>-0.06 ***</td>
<td>-0.05 ***</td>
</tr>
<tr>
<td>Power Ratio^2</td>
<td>0.00 ***</td>
<td>0.00 ***</td>
</tr>
<tr>
<td>Iteration Group</td>
<td></td>
<td>-0.08 ***</td>
</tr>
<tr>
<td>Power</td>
<td>-0.81 ***</td>
<td>-0.67 ***</td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>-2.83 ***</td>
<td>-2.22 ***</td>
</tr>
<tr>
<td>War Cost Maximum</td>
<td>1.13 ***</td>
<td>0.83 ***</td>
</tr>
<tr>
<td>Disproportionate War Cost</td>
<td>0.55 ***</td>
<td>0.42 ***</td>
</tr>
<tr>
<td>Reparations</td>
<td>-0.38 ***</td>
<td>-0.31 ***</td>
</tr>
<tr>
<td>Metabolism</td>
<td>0.01 *</td>
<td>0.01 **</td>
</tr>
<tr>
<td>Constant</td>
<td>0.49 ***</td>
<td>-0.55 ***</td>
</tr>
<tr>
<td>Observations</td>
<td>2,857,748</td>
<td>2,857,748</td>
</tr>
<tr>
<td>Pseudo R^2</td>
<td>0.14</td>
<td>0.15</td>
</tr>
</tbody>
</table>

(*** p < 0.01; ** p < 0.05; * p < 0.10)

Table 5.12: Impact of Power Ratio
However, as we have seen before, the difference in the average of war probabilities is so big between dyad types that we have to investigate it in detail. I ran probit analyses on each dyad type separately in order to compare the impact of the power ratio. Tables 5.13, 5.14, 5.15 and 5.16 show the results of those analyses. In Table 5.13 and Table 5.14, we can see that the coefficients of power ratio are negative (−0.01 and −0.02, respectively). On the other hand, in Table 5.15 and Table 5.16, the coefficients of power ratio become positive (2.47 and 0.22, respectively). In case of the minor-major dyads, the high value of coefficient means that a minor state tends to initiate a war more likely against a major state, as its power ratio increases relative to its major target.43 Although the (non-major) initiator is weaker than her major target, there are many cases that induce her to initiate a war. In these cases, the increase of the power ratio between the initiator and the target (that must be less than 1.0) tends to impact on the probability of war more in the minor-major dyad than in other dyads. Thus we can find support for Proposition 7 only in the dyads composed of minor states or in the dyads composed of a minor state (as an initiator) and a major state (as a target). In other cases, the result does not show that a higher power ratio leads to a higher war ratio with their negative coefficients. This implies that major powers are more prudent in initiating war than minor states (maybe due to the high cost of war), regardless of the size of the target. As such, along with the case of economic interdependence, the type of dyads also has an impact on the way the power ratio affects the probability of war.

---

43 This may happen when the two states' power levels are almost the same, even though the minor state chooses a war against her major partner. In addition, due to the factor of power estimation error, it is very likely for a minor state to initiate a war if its power ratio increases relative to a major partner, whose power level is underestimated by the initiator.
### Table 5.13: Impact of Power Ratio in Major-Major Dyads

<table>
<thead>
<tr>
<th>Variables</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Ratio</td>
<td>-0.01***</td>
</tr>
<tr>
<td>Power</td>
<td>-0.91***</td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>-1.79***</td>
</tr>
<tr>
<td>War Cost Maximum</td>
<td>0.37***</td>
</tr>
<tr>
<td>Disproportionate War Cost</td>
<td>0.60***</td>
</tr>
<tr>
<td>Reparations</td>
<td>-0.15***</td>
</tr>
<tr>
<td>Metabolism</td>
<td>0.03**</td>
</tr>
<tr>
<td>Constant</td>
<td>-0.11***</td>
</tr>
<tr>
<td>Observations</td>
<td>461,444</td>
</tr>
<tr>
<td>Pseudo R²</td>
<td>0.02</td>
</tr>
</tbody>
</table>

\[ (*** \ p < 0.01; ** \ p < 0.05; * \ p < 0.10) \]

### Table 5.14: Impact of Power Ratio in Major-Minor Dyads

<table>
<thead>
<tr>
<th>Variables</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Ratio</td>
<td>-0.02***</td>
</tr>
<tr>
<td>Power</td>
<td>-0.46***</td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>-2.88***</td>
</tr>
<tr>
<td>War Cost Maximum</td>
<td>1.69***</td>
</tr>
<tr>
<td>Disproportionate War Cost</td>
<td>0.76***</td>
</tr>
<tr>
<td>Reparations</td>
<td>-0.14***</td>
</tr>
<tr>
<td>Metabolism</td>
<td>0.00</td>
</tr>
<tr>
<td>Constant</td>
<td>-0.06***</td>
</tr>
<tr>
<td>Observations</td>
<td>1,651,854</td>
</tr>
<tr>
<td>Pseudo R²</td>
<td>0.15</td>
</tr>
</tbody>
</table>

\[ (*** \ p < 0.01; ** \ p < 0.05; * \ p < 0.10) \]
<table>
<thead>
<tr>
<th>Variables</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Ratio</td>
<td>2.47***</td>
</tr>
<tr>
<td>Power</td>
<td>-0.47***</td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>2.90***</td>
</tr>
<tr>
<td>War Cost Maximum</td>
<td>-2.55***</td>
</tr>
<tr>
<td>Disproportionate War Cost</td>
<td>0.21**</td>
</tr>
<tr>
<td>Reparations</td>
<td>-0.39*</td>
</tr>
<tr>
<td>Metabolism</td>
<td>0.19**</td>
</tr>
<tr>
<td>Constant</td>
<td>-4.17***</td>
</tr>
<tr>
<td>Observations</td>
<td>106,959</td>
</tr>
<tr>
<td>Pseudo $R^2$</td>
<td>0.18</td>
</tr>
</tbody>
</table>

(*** $p < 0.01$; ** $p < 0.05$; * $p < 0.10$)

Table 5.15: Impact of Power Ratio in Minor-Major Dyads

<table>
<thead>
<tr>
<th>Variables</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Ratio</td>
<td>0.22***</td>
</tr>
<tr>
<td>Power</td>
<td>-0.95***</td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>-3.21***</td>
</tr>
<tr>
<td>War Cost Maximum</td>
<td>1.29***</td>
</tr>
<tr>
<td>Disproportionate War Cost</td>
<td>0.17***</td>
</tr>
<tr>
<td>Reparations</td>
<td>-0.95**</td>
</tr>
<tr>
<td>Metabolism</td>
<td>0.01</td>
</tr>
<tr>
<td>Constant</td>
<td>0.68***</td>
</tr>
<tr>
<td>Observations</td>
<td>637,221</td>
</tr>
<tr>
<td>Pseudo $R^2$</td>
<td>0.04</td>
</tr>
</tbody>
</table>

(*** $p < 0.01$; ** $p < 0.05$; * $p < 0.10$)

Table 5.16: Impact of Power Ratio in Minor-Minor Dyads
The impact of dyadic properties—economic interdependence and heterogeneity in state size (or capabilities)—on the probability of war was explored in this chapter. The level of economic interdependence in a dyad seems to be a very significant factor in reducing the probability of war. In particular, the average of war probability is around 20 percent between the interdependence levels 0.0 and 0.4. Beyond this range, war seldom happened. The types of dyads—major to major, major to minor, minor to major, and minor to minor—showed different averages of war probabilities, 22.9, 16.0, 0.9 and 50.9, respectively. Also the impact of economic interdependence on the probability of war was different across different types of dyads.

As such, the impact of economic interdependence on the probability of war is strongest in the minor-minor dyads, while it is weakest in the major-minor dyads. Of course, we can say that economic interdependence decreases the probability of war in dyads and that the impacts seem different across various dyad types. In the case of the major-minor dyads, it seems that a major state as the initiator of interaction considers the cost of war, which reduces the expected utility of war against a minor state (as the target). According to the war module used in Cusack and Stoll’s original model, the cost of war increases according to the size of the initiator’s power regardless of the target’s power.44 On the other hand, minor states seem not to be restrained by this condition; a cell

---

44 The variable of disproportionate war cost puts limits on this: A stronger state (or alliance) in a war has to pay a certain proportion of her power, even though her war cost ratio is less than that of her weaker partner.
(territory) with its resources may be a big gain for a small state in contrast to the cost of war and indemnities if she loses the war.

The distribution of war across different types of dyads is also investigated using two-sample t-tests. For Proposition 5, two samples – the major-major dyads and the minor-minor dyads – were selected from the whole 2.85 million samples. The averages of both groups, 0.23 for the major-major dyad and 0.51 for the minor-minor dyad, are significantly different, so that we can say that wars between major powers are less frequent than those between minor powers. For Proposition 6, I used the total 2.85 million cases again, and they were re-classified into the homogeneous dyad (the major-major dyad and the minor-minor dyad) and the heterogeneous dyad (the major-minor dyad and the minor-major dyad). The latter group shows a mean of 0.15 in its war probability, while the former group shows a mean of 0.39. Based upon the t-test, we have to reject Proposition 7. Wars tend to happen more frequently if the dyad is composed of homogeneous states in terms of state power.

For Proposition 7 about the impact of power ratio (as a continuous variable) on the war probability, I ran a probit analysis again for the total sample. The probability of war decreases as the power ratio between the initiator and the target increases, so that it shows a consistent pattern with the previous one. The impact of the iteration group also has a negative impact on war frequencies, in addition to the power ratio variable. Particularly, considering the difference between types of dyad, separate probit analyses were conducted on them. The results show that, in the major-major dyad (−0.01) and the major-minor dyad (−0.02), the impact of power ratio on war probability is negative and significant, even though its amount is not so big. In contrast, the minor-major dyad (2.47)
and the minor-minor dyad (0.22) show positive impacts by power ratio on the probability of war. Thus the test of Proposition 7 shows a mixed result – partly positive and partly negative impacts on the frequency of war by the power ratio between the initiator and the target, according to the type of dyads.
CHAPTER 6

POLARITY, POWER CONCENTRATION AND WAR

6.1 Research Themes

The structural theory of Waltz has been a mainstream in explaining war. One of the major purposes of this thesis is to contrast this neorealist structural approach to the bottom-up framework used in this study. I would like to explore how his theory works in the mixed model of security and trade examined in this study. Also I would like to find any compromise between the two approaches, if Waltz's theory matters. The structural neorealist theory explains the war/conflict dynamics with structural factors, while this thesis focuses on the micro-level interactions with the addition of the trade option. By changing the values of parameters used in modeling the virtual worlds simulated, the roles of individual-level and the dyad-level factors have been investigated in the previous chapters. This chapter focuses on the system-level aspects by inserting the factors of polarity and power concentration among major powers into the equations, so that we can figure out whether and how much the systemic factors influence the probability of war. If
we find any significant patterns, then we might have to compromise between the
Waltzian theory of international relations and the newly introduced CAS-based, bottom-
up style approach.

The first structural factor to be tested is polarity. Many international relations
scholars have discussed the influence of polarity since Waltz's suggestion. Some have
emphasized the positive relationship between polarity and conflict, while others argued
for a negative relationship between them. As far as this theme is concerned, this chapter
tries to test Waltz's main argument that the world will be more peaceful with more
polarity (i.e., with fewer major powers). Proposition 8 expresses this theme in a simple
form. The measurement of polarity, however, follows a more complex procedure based
upon empirical data which will be discussed in the research design section in detail. In
addition to this simple form of the proposition, another test will be added to this chapter,
with the variable of the number of major powers, in order to see whether there is any
difference in expressing Proposition 8 with the variable polarity and with the variable of
major power number:

Proposition 8: The probability of war declines as the number of poles
decreases.

On the other hand, it is worth analyzing the effects of another structural
dimension – power concentration among major powers – which has been suggested to
replace the simple notion of polarity in explaining conflict. Edward Mansfield has argued
that the analysis of polarity in international relations is founded upon two problematic
assumptions; first, poles are structurally equivalent or inequality among major powers is not important; second, major powers that are not poles have no influence on the distribution of power and the pattern of conflict (Mansfield, 1994, 72). Thus he proposed to use Ray and Singer’s index of power concentration to measure the inequality among themselves upon the assumption that the inequality among themselves may have impacts on war frequencies. His finding out of empirical data was the inverted U-shaped distribution of war; at the lower and higher levels of power concentration, the frequency of war is low; at the intermediate level of power concentration, the frequency of war is high. This is a challenging pattern to explain in international relations, but it seems that there has not been any persuasive theory about this pattern. Although we do not have any dominating theories about this phenomenon, I would like to test this theme in my simulated world with Proposition 9:

Proposition 9: The probability of war follows an inverted U-shaped curve as the level of power concentration increases among major states.

6.2 Research Design

6.2.1 Variables

The most important thing to discuss regarding research design of this chapter is the concept and measurement of polarity. Although political scientists have used this term in describing the structural property of world politics, few have suggested any
operationalizable definition of it. There seems not to be any empirical studies that work as a guideline for this problem. Although the COW dataset marks clearly which countries had or have been “major,” it does not classify periods in terms of polarity, either. Considering this point, I would explore the COW dataset to find any empirical clues for classifying the “polarity” of world structure. One thing to mention is that we have applied this term for only two historical cases – the multipolarity in the 19th century and the bipolarity in the post-Cold War period. Therefore, I would search for any substantive patterns from COW by investigating these periods upon some combination of conditions, especially the configuration of major powers’ capabilities.

Upon the collection of tangible patterns from the real-world data, I would find some reasonable criteria for delineating “unipolar” cases (which might not have existed in history) and “apolar” cases (which means that the structure is not polar or cannot be classified as unipolar, bipolar, or multipolar) as well as the familiar structural patterns of “bipolar” and “multipolar.” While this classification may not be included in existing theories, we need this kind of rigorous formal classification for our simulated worlds. Focus will be put on the power relationship among major powers, following the COW guideline and Waltz’s theory. Figure 6.1 shows how each system is classified regarding its polarity, with the composition of major powers.
Figure 6.1: System Polarity Classification Algorithm
Figure 6.2: Power Configuration of Major States, 1816-1965
In Figure 6.1, the first work for the classification is finding all major states in the system. As mentioned before, "major" states are identified if their relative power shares exceed 3 percent, but the total number of major powers is limited to eight (as Waltz suggested and the empirical COW dataset shows). In particular, based upon the COW dataset, I would classify the system "unipolar" if the biggest major power holds more than 40 percent of world power which has been rarely achieved by a single country in history. If the second strongest state holds more than 30 percent of world power, then the system is marked as "bipolar." In the case when no state holds more than 40 percent of world power, if the sum of capabilities between the strongest power and the second strongest power exceeds 50 percent but their difference is less than 10 percent, the system is also classified as "bipolar." If their difference is more than 10 percent, the system is identified as "multipolar" as far as there are three or more major states in the world. Otherwise, the system is coded as "apolar." Based upon the analysis of the 19th-century empirical data, the system is classified as "multipolar" if the sum of capabilities of all the major powers' power is over 70 percent under the condition that there are three or more major powers. Otherwise, the system is classified as "apolar," either. Of course, the system with no major power is coded as "apolar" which represents many initial conditions of simulation runs.

This classification algorithm follows the empirical trend analysis of the COW dataset, because we do not have any articulated theoretical bases. Figure 6.2 shows some patterns of major power configurations for the past 150 years. The top line is the sum of major powers' capabilities, measured by the sum of relative shares across six different sub-indices. The sum of five to eight major countries in this 150-year period is between 179.
60 and 80 percent. So I set the bottom line of classifying multiplicity to 70 percent of world power, considering some possible over-evaluation of major powers due to the difficulty in data collection for small countries. The empirical analysis also finds that the strongest power's capability has rarely exceeded 30 percent during the last 150 years. Thus we may call the system if the strongest power's share is over 40 percent, which has never been reached by a country in history but must be logically possible in our virtual worlds. One noticeable pattern in the graph is the difference between the top line (representing the sum of major powers' capabilities) and the third line (the two strongest powers' sum). Since the Cold War, it has narrowed down and we call this period "bipolar." So I would call the system "bipolar" if the two strongest countries' capabilities share exceeds 50 percent (a little bit higher than the empirical trend) but their difference is less than 10 percent. As such, the system polarity used in this simulation is based on the analysis of historical cases – bipolarity and multipolarity – and its extensions to the unipolarity and apolarity in the virtual world.

Another structural variable that is used in this chapter is the power concentration among major powers. Mansfield used Ray and Singer's index that measures asymmetry in the distribution of power among major states (Mansfield, 1994, 72):

\[
CON_i = \sqrt{\frac{\sum_{n=1}^{N_i} (S_n) - \frac{1}{N_i}}{1 - \frac{1}{N_i}}}
\]
where $S_n$ measures the proportion of each major power’s capabilities among the total of major powers’ capabilities at time $t$, and $N_t$ is the number of major powers in the system at time $t$. As Mansfield explains, this $CON$ index is a “continuous” variable ranging from zero to one and measures the inequality of capabilities among all the major powers in the system.\textsuperscript{44} For example, if one major state holds most power among other major states, then the value of $CON$ is close to 1.0; and if several major states share the same amount of power, then the value of $CON$ is zero (\textit{Ibid.}, 72-73).\textsuperscript{45} The first is the case of extreme asymmetry of power, while the latter is an extreme symmetry of power among major powers. Figure 6.3 shows the historical distribution of $CON$ values among major powers, measured by this method. We can observe that the $CON$ values for the whole period fluctuate between 0.15 and 0.5, except some exceptional time points.\textsuperscript{46} Mostly the value is between 0.2 and 0.4. Another feature from Figure 6.3 is that the value of $CON$ increases from the period of multipolarity in the 19\textsuperscript{th} century to the period of bipolarity since 1945. We do not have any other empirical data before and after the period, but the simulation runs will generate a virtual dataset even for the unipolar and apolar cases that have not yet existed in the period of which data are available to us.

\textsuperscript{44} This formula for the $CON$ is not applicable to the condition when there is only one major state in the world.

\textsuperscript{45} If there is only one state in the world (i.e., a unique state actor in the world), then this $CON$ index cannot be identified. This index assumes the existence of multiple state agents in the world.

\textsuperscript{46} Mansfield’s sample data for some selected years show similar patterns. The four years’ $CON$ values, from the same source of COW dataset, are 0.222 (1840), 0.259 (1855), 0.326 (1920) and 0.319 (1955) for five major states. Curiously, there seems to be some slight differences in each year’s average $CON$ values between Mansfield’s and my analysis. It is not clear why it happens, but a suspect is the difference in the summing-up process from the six sub-indices of state power. However, the difference seems not significant and the process of $CON$ calculation is exactly the same.
6.2.2 Equations

For Proposition 8, which mentions the relationship between polarity and the probability of war, I develop two equations as I want to see two dimensions of polarity at the same time. The first is with the dummy variables for each type of polarity – unipolarity, bipolarity, and multipolarity. The case of apolarity was used for the base model for these dummy variables. The second equation is with the number of major powers in the system; this equation is an additional one for Proposition 8 because the variable of polarity is not so sophisticated with its four-tier classification and because the types of polarity are not based on clear definitions. The number of major powers will range between 0 and 8, according to the basic configuration of the simulation rules. Also this variable is clear in its meaning when we interpret the result. As such, Proposition 8 will be tested with the probit analysis using the following two equations.

**Proposition 8: The probability of war declines as the number of poles decreases.**

\[
Pr(War) = \beta_0 + \beta_1 \cdot \text{multipolarity} + \beta_2 \cdot \text{bipolarity} + \beta_3 \cdot \text{unipolarity} + \beta_4 \cdot \text{power}_{sd} + \beta_5 \cdot \text{error}_{sd} + \beta_6 \cdot \text{warcost} + \beta_7 \cdot \text{wcdispar} + \beta_8 \cdot \text{repar} + \beta_9 \cdot \text{meta}_{sd}
\]

\[
Pr(War) = \beta_0 + \beta_1 \cdot \text{whether major} + \beta_2 \cdot \text{num major} + \beta_3 \cdot \text{num major}^2 + \beta_4 \cdot \text{power}_{sd} + \beta_5 \cdot \text{error}_{sd} + \beta_6 \cdot \text{warcost} + \beta_7 \cdot \text{wcdispar} + \beta_8 \cdot \text{repar} + \beta_9 \cdot \text{meta}_{sd}
\]

(where **multipolarity** is the dummy variable for multipolarity, **bipolarity** is the dummy variable for bipolarity,
unipolarity is the dummy variable for unipolarity, whether major is the dummy variable for any major power, num_major is the number of major powers in the system, num_major\(^2\) is a squared term of num_major, pow_sd is the standard deviation of power distribution, error_sd is the standard deviation of power estimation error, warcost is the maximum proportion of power to lose in a war, wcdispar is the prewar power ratio to asymmetric war cost, repar is the proportion of the loser's power given to winners, and meta_sd is the standard deviation of metabolism distribution.

For Proposition 9, which mentions the relationship between CON and the probability of war, I will use the following equation, including the CON variable. As Proposition 9 deals with a curvilinear relationship between power concentration and war probability, I will insert a squared term for the power concentration among major powers. The equation will be accompanied by some graphical illustrations that show the pattern of curvilinear relationships.

Proposition 9: The probability of war follows an inverted U-shaped curve as the level of power concentration increases among major states.

\[
War = \beta_0 + \beta_1 \cdot CON + \beta_2 \cdot CON^2 + \beta_3 \cdot pow_sd + \beta_4 \cdot error_sd
+ \beta_5 \cdot warcost + \beta_6 \cdot wcdispar + \beta_7 \cdot repar + \beta_8 \cdot meta_sd
\]

(where CON is the level of capability concentration among major powers, CON\(^2\) is a squared term of CON, pow_sd is the standard deviation of power distribution, error_sd is the standard deviation of power estimation error, warcost is the maximum proportion of power to lose in a war, wcdispar is the prewar power ratio to asymmetric war cost, repar is the proportion of the loser's power given to winners, and meta_sd is the standard deviation of metabolism distribution)
6.3 Analysis of Results

6.3.1 Polarity and War

A new simulation of 1,000 runs was conducted for this chapter, which generated a total of 559,619 appropriate cases of dyadic interaction. Of the whole cases, more than 40 percent happened in apolar systems and about 35 percent in unipolar systems. Multipolar and bipolar systems show around 10 percent cases of interacting dyads for each. Figure 6.4 shows the distribution of cases upon polarity among the whole cases. Table 6.1 shows the probability of war across these polarity sets. The average probability of war across polarity sets is 24.93, so roughly one out of four dyads ended in the occurrence of war. However, there is a difference – the probability of war is highest (37.64) if there is no poles in the system, while lowest (11.67) in the unipolar system. On the other hand, war probability is at the intermediate level for the multipolar system (26.00) and the bipolar systems (15.01). Thus we can conclude from this observation that war happens less frequently as the number of poles decreases. The nature of inter-state relations in this artificial world tends to be more peaceful with the increase of polarity in world politics, so they are consistent with Waltz’s argument that a bipolar system is more peaceful than a multipolar world as well as an extension of his logic to the unipolar world. As he put it, the “smaller is more beautiful than small” (1979, 134).

47 I removed these 28 cases from the analysis because they cannot be classified upon the criterion of polarity due to the lack of both partners in dyads. These cases can happen, even though they are rare, when there is no state holding positive expected utility.
Figure 6.4: Distribution of Polarity

<table>
<thead>
<tr>
<th>Polarity</th>
<th>No War (%)</th>
<th>War (%)</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apolar</td>
<td>62.36</td>
<td>37.64</td>
<td>243,125</td>
</tr>
<tr>
<td>Multipolar</td>
<td>74.00</td>
<td>26.00</td>
<td>65,443</td>
</tr>
<tr>
<td>Bipolar</td>
<td>84.99</td>
<td>15.01</td>
<td>49,953</td>
</tr>
<tr>
<td>Unipolar</td>
<td>88.33</td>
<td>11.67</td>
<td>201,098</td>
</tr>
<tr>
<td>Total</td>
<td>75.07</td>
<td>24.93</td>
<td>559,619</td>
</tr>
</tbody>
</table>

Table 6.1: Polarity and the Probability of War
<table>
<thead>
<tr>
<th>Variables</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multipolarity</td>
<td>-0.32***</td>
</tr>
<tr>
<td>Bipolarity</td>
<td>-0.74***</td>
</tr>
<tr>
<td>Unipolarity</td>
<td>-0.90***</td>
</tr>
<tr>
<td>Power</td>
<td>-0.49***</td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>-2.24***</td>
</tr>
<tr>
<td>War Cost Maximum</td>
<td>1.35***</td>
</tr>
<tr>
<td>Disproportionate War Cost</td>
<td>0.50***</td>
</tr>
<tr>
<td>Reparations</td>
<td>-0.27***</td>
</tr>
<tr>
<td>Metabolism</td>
<td>0.07***</td>
</tr>
<tr>
<td>Constant</td>
<td>0.13***</td>
</tr>
<tr>
<td>Observations</td>
<td>559,619</td>
</tr>
<tr>
<td>Pseudo R²</td>
<td>0.08</td>
</tr>
</tbody>
</table>

(*** p < 0.01; ** p < 0.05; * p < 0.10)

Testing Proposition 8: The probability of war declines as the number of poles decreases.

T-test for the difference between coefficients:

- **H₀ (Multipolarity = Bipolarity):** \( t = 45.68, P > |t| = 0.0000 \)
- **H₀ (Bipolarity = Unipolarity):** \( t = -329.65, P > |t| = 0.0000 \)

**Table 6.2: Impact of Polarity on the Probability of War**

If we run a probit regression for this relationship with the dummy variables for different types of polarity, then the negative impact of the sharper polarity on the probability of war becomes clearer. Table 6.2 summarizes the result. The coefficients of different polarities increase to the negative direction from the base model (with apolarity) to multipolarity \((-0.32)\), to bipolarity \((-0.74)\), and finally to unipolarity \((-0.90)\). These coefficients are significantly different from each other as the t-test results show. This indicates that we tend to have a more peaceful world with the increase of polarity (i.e., with fewer poles) at least in the artificial world constructed by the major international relations theories, like that in this study. Thus we find clear support for Proposition 8 that
the probability of war becomes less with the increase of polarity in the system. Here, considering that the levels of polarity are only four, we may need to expand them by incorporating the number of major powers in the system, as discussed in the research design section. Because the measurement of polarity does not always reflect the whole group of major powers, it may be reasonable to use the number of major powers instead of the simple index of polarity. Let me explain Figure 6.5, Table 6.3, and Figure 6.6.

Figure 6.5 shows the distribution of the number of major states between the minimum zero and the maximum eight. More than 35 percent of cases represent the case of eight major states, while other cases from zero to seven are distributed around or below 10 percent. Table 6.3 and Figure 6.6 show the average probability of war across different numbers of major state. In Table 6.3, we can see that the probability of war is the lowest when the number of major powers is two or three, 17.47 and 17.09, respectively. Considering that the average probability of war across the numbers of major powers is 24.93, we can say that wars happen less frequently with a smaller number of major powers in the world. The above graph and this table make us to believe in any nonlinear pattern of the likelihood of war, so I included a squared term for the number of major states in my probit analysis of Table 6.4. An exceptional is that the probability of war gets higher a little bit if the number of major states is reduced to one from two or three. This is a curious phenomenon, but it might be related to the large variance in the size of the only major powers.48

48 While there are many cases of the only major power in the world, such as unipolarity in which the major power is a big empire, there also are other cases in which a small major power (just over three percent of the world total power) emerges out of the competition of very many small states. That is, the case of only one major power in this analysis represents a superset of the unipolarity case. This may make the probability of war higher than that in the cases of two or three major states.
Figure 6.5: Distribution of the Number of Major States

<table>
<thead>
<tr>
<th>Number of Major States</th>
<th>No War (%)</th>
<th>War (%)</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>41.00</td>
<td>59.00</td>
<td>12,568</td>
</tr>
<tr>
<td>1</td>
<td>78.59</td>
<td>21.41</td>
<td>42,354</td>
</tr>
<tr>
<td>2</td>
<td>83.53</td>
<td>17.47</td>
<td>53,197</td>
</tr>
<tr>
<td>3</td>
<td>82.91</td>
<td>17.09</td>
<td>60,714</td>
</tr>
<tr>
<td>4</td>
<td>81.35</td>
<td>18.65</td>
<td>60,413</td>
</tr>
<tr>
<td>5</td>
<td>78.40</td>
<td>21.60</td>
<td>50,819</td>
</tr>
<tr>
<td>6</td>
<td>74.92</td>
<td>25.08</td>
<td>40,947</td>
</tr>
<tr>
<td>7</td>
<td>71.87</td>
<td>28.13</td>
<td>36,019</td>
</tr>
<tr>
<td>8</td>
<td>70.04</td>
<td>29.96</td>
<td>202,588</td>
</tr>
<tr>
<td>Total</td>
<td>75.07</td>
<td>24.93</td>
<td>559,619</td>
</tr>
</tbody>
</table>

Table 6.3: Number of Major States and the Probabilities of Trade and War
Category 9 actually means zero in the number of major powers. It is arbitrarily recoded to 9, after the maximum number 8, in order to compare it with the scale of polarity.

Figure 6.6: Number of Major States and the Probability of War
<table>
<thead>
<tr>
<th>Variables</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whether Major Power</td>
<td>-1.11***</td>
</tr>
<tr>
<td>Number of Major States</td>
<td>-0.05***</td>
</tr>
<tr>
<td>Number of Major States$^2$</td>
<td>0.01***</td>
</tr>
<tr>
<td>Power</td>
<td>-0.50***</td>
</tr>
<tr>
<td>Power Estimation Error</td>
<td>-2.20***</td>
</tr>
<tr>
<td>War Cost Maximum</td>
<td>1.95***</td>
</tr>
<tr>
<td>Disproportionate War Cost</td>
<td>0.53***</td>
</tr>
<tr>
<td>Reparations</td>
<td>-0.11***</td>
</tr>
<tr>
<td>Metabolism</td>
<td>-0.06***</td>
</tr>
<tr>
<td>Constant</td>
<td>0.55***</td>
</tr>
<tr>
<td>Observations</td>
<td>559,619</td>
</tr>
<tr>
<td>Pseudo R$^2$</td>
<td>0.04</td>
</tr>
</tbody>
</table>

(*** $p < 0.01$; ** $p < 0.05$; * $p < 0.10$)

Table 6.4: Number of Major States and the Probability of War

Table 6.4 is the result of a probit analysis with the number of major states as an explanatory variable. The existence of any major power decreases the probability of war in the system, as the coefficient of the variable "whether major power" is negative. The coefficient of the number of major states (N) is -0.05 but its squared term is 0.01 which is significant. Thus we can say that the relationship between the number of major powers and the probability of war is curvilinear as we see in Figure 6.6. Also we had better use the index of polarity in parallel with the number of major states in order to interpret this result, as both indices are complementary with each other in describing structural features. As such, the impact of polarity (and also the number of major states in the system) on the probability of war is verified here so that we cannot ignore the Waltzian structural
neorealism in explaining the dynamics of world politics. Detailed investigation with another variable – power concentration – will follow this section, but it is worth exploring the influence of polarity a little bit further.

Figure 6.7 and Figure 6.8 illustrate how polarity indirectly affects the way other variables – economic interdependence and power ratio – influence the probability of war. It seems that whether the system is bipolar, multipolar, or unipolar makes differences in assessing the probability of war with other factors. Figure 6.7 shows the impacts of economic interdependence on war probability, which we already investigated in Chapter 5. Each curve represents the relationship between economic interdependence and war probability, and distinguishes itself from others. The role of economic interdependence is highest in the multipolar world, while it is lowest in the unipolar world. It is interesting that the unipolar system is much less influenced by the level of interdependence.

The role of polarity in its indirect influence seems a bit different in case of power ratio. Figure 6.8 illustrates the relationship between power ratio (between two countries in a dyad) and war probability by each polarity group. Here, the impact of power ratio is greatest in the bipolar world, while it is lowest in the unipolar world where it goes down slowly as the power ratio increases. These results imply that multipolar worlds are relatively more sensitive to the potential benefits from trades rather than from conflicts than bipolar and unipolar worlds are. On the other hand, the bipolar system seems to be the most sensitive to the power inequality among interactive partners in its dynamics. As such, the structural variable of polarity has big impacts on war and conflict frequency both directly and indirectly.
Figure 6.7: Interdependence and the Probability of War by Polarity
Figure 6.8: Power Ratio and the Probability of War by Polarity
6.3.2 Power Concentration and War

The introduction of power concentration seems reasonable in terms of the level of sophistication, which starts from the simple scheme of polarity, in analyzing the structural factors. That is, after investigating the roles of polarity and the number of major powers, dealing with power concentration must be the next step because of its comprehension of both the number of major powers and the inequality among themselves. However, due to the intrinsic limitation of the concept itself (the inequality "among major powers" and the need for the system to be one of three major polarities), I have to select the cases that hold more than one major powers. The total number of cases that fit this scheme, after deleting 276,679 inappropriate cases, reaches 282,940. Among all the remaining cases, about 60 percent are unipolar cases, 18 percent are multipolar cases, and 23 percent are bipolar cases. As there have been many cases of N = 8 that are not to be classified as "multipolar" because the sum of their capabilities does not sum up to 70 percent, it is reasonable to remove them in analyzing the impact of power concentration among major powers. Therefore, the size of the dataset for the analysis of this section is dramatically reduced to 1.6 million upon a stricter requirement of selection – both polarity and the number of major powers. Figure 6.9 redraws Figure 6.5 with the new dataset, in which we can observe the remarkable decrease of N = 8 cases.

49 The deleted cases are those including no major powers or only one major power, to which we cannot apply the formula used by Mansfield. Due to the nature of this formula – i.e., assuming the existence of at least two major powers – it does not make sense to discuss power concentration without two or more major states in the world.

195

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 6.9: Distribution of the Number of Major States (N > 1)

Figure 6.10: Power Concentration among Major States
Figure 6.10 shows the distribution of power concentration values among major powers in the simulated dataset. In this artificial world, the CON values are distributed between 0.5 and 1.5. Although the real-world data shows the distribution of CON values with the narrow range between 0.2 and 0.4 (and at least below its maximum around 0.5), this simulated dataset covers the cases over 0.5 which are not available in the real-world dataset. Roughly half of the cases show the CON values over 0.5. The average value of the probability of war is 15.47, but the variation is quite big across different values of CON. Figure 6.11 is the distribution of war probabilities across (collapsed and recoded) power concentration among major powers. War happens at the rate of 20~30 percent when power concentration level is between 0.1 and 0.3, but it drops down below 15 percent if the concentration level is over 0.4.

The result of Figure 6.11 seems to be a challenging phenomenon for major international relations theories, because it has a close relationship to Proposition 9 which criticizes the simple use of polarity in analyzing international conflicts. Mansfield has argued for the CON index and the pattern of inverted-U shape in the probability of war across the CON spectrum. His argument applies to the artificial worlds in this study, either, but not in the whole range of the CON values. First of all, the inverted-U shape is shown between zero and 0.4; over that value, the overall pattern slowly drops down. Second, around 0.5 of the CON value, the probability of war smoothly rises again. However, the overall pattern of the inverted-U shape between the CON among major powers and the probability of war happens in this artificial world, like in the case of the real-world dataset.
Figure 6.11: Power Concentration and the Probability of War

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Table 6.5: Impact of Power Concentration on War Probability

Mansfield's observations were limited to less than 200 years of period, so that the result in this chapter may help in developing theory further in explaining conflict patterns in international relations. Anyhow, in Figure 6.11, the probability of war is highest at the low $CON$ values (between 0.1 and 0.3) so that we could observe an inverted-U shape between the $CON$ index and the likelihood of war at lower $CON$s. As the value of $CON^2$ is significant, we can prove that the relationship is not linear. With the exception of the cases between $CON = 0.5$ and 0.6 where the probability of war slightly increases, the world becomes more peaceful with the increase of power concentration among major powers. As such, we can say that the overall pattern of the probability of war by the impact of the power concentration among major powers takes an inverted-U shape as expressed in Proposition 9.
6.4 Summary

The effects of structural variables, in contrast to the fundamental bottom-up philosophy of this study, are more important in the artificial world than expected. The first variable, polarity, which was measured upon some empirical bases, has a great influence on the probability of war. As the structural neorealists argued, the more polar the world is, the more peaceful it becomes. Only 11.67 percent cases of unipolar worlds were marked as at war, while 15.01 percent of bipolar worlds and 26.00 percent of multipolar worlds recorded wars. The worlds classified as “apolar,” whether there is no poles or the system cannot be classified by any known type of polarity, recorded 37.64 percent of war cases. A more sophisticated analysis was done with the number of major states as a main explanatory variable. While polarity has a directly linear relationship with the probability of war, the number of major states (measured by the power share over 3 percent, up to eight as its maximum in the system) shows a little bit complicated pattern. For example, the probability of war was lowest when the number of major states is 2 or 3. In these cases, war happens only at the ratio of 17.47 percent and 17.09 percent, respectively. On the other hand, when there is only one major power in the system, the probability of war was higher (21.41). The overall trend was a slow increase of the probability of war with the increase of the number of major states. The factors of economic interdependence and power ratio have different influences on the probability of war across different types of polarity. For example, economic interdependence plays the strongest roles in the multipolar system, while power ratio does so in the bipolar system.
The introduction of power concentration among major states is an improved way to investigate the role of structural factors in explaining international conflict. Throughout all cases, power concentration ratios ($CON$) among major powers are distributed over the range of 0.0 and 1.0. This kind of distribution over the whole range seems to be an advantage of computer simulations because previous literature could find the concentration pattern mostly between 0.2 and 0.4. The addition of extra cases from this artificial world will definitely help to improve our structural theory of conflict. The probability of war across different $CON$ values was 15.47 on average, but its variation need to be mentioned. Particularly, war happens most frequently when $CON$ is between 0.1 and 0.3 (at the rate between 0.2 and 0.3), while it slowly decreases with the increase of $CON$. The existing empirical analysis shows a similar result with this study, especially mostly at the lower portion of $CON$ values. As Mansfield argued, an overall pattern of the inverted U-shape relationship happened between $CON$ and the probability of war even in this artificial world, despite a couple of exceptions.

As such, we have found that structural factors – polarity, the number of major states, and the power concentration among major states – play important roles in determining the patterns of war in the artificial world politics. Most observations seem not to diverge much from the real-world pattern, so that we could confirm the reasonableness of existing major theories that were implemented in this simulation study and we could get some clues to the future scenarios of world politics we have not yet experienced. For example, in the unipolar world, we may expect a more peaceful world relative to the bipolar or the multipolar worlds. However, just one major state in the world seems not to be the best scenario for our peace: in terms of the number of major
powers, two or three are the best for the maintenance of peaceful worlds. The combination of these two conditions – two or three major powers in a unipolar world – may be the best option for world peace, and we may need a theory about this in the future.

In sum, structural factors are influential on the conflict patterns in the artificial worlds on the computer screen. This point is important in two ways. First, this result has some effects to verify the logical persuasiveness of the Waltzian structural neorealism. Despite many probable weaknesses and criticisms, it seems that structural factors should not be ignored in analyzing world politics as Waltz recommended. Second, it urges us to accelerate our efforts to develop a kind of combined theory based both on the bottom-up and the top-down approaches. As I discussed in the introductory chapter, Waltz’s approach takes an extreme top-down way, while the basic philosophy of this study stands for a bottom-up method. Despite a huge gap between these two approaches, I do not assume that both are incompatible at all. Perhaps we can compromise between them with various options. For instance, we may understand the working of structural factors as internal (or circumstantial) rather than as external (or given) to the system. Any theoretical development about this point goes beyond this study, even though the simulation results have many clues for this kind of compromise.
CHAPTER 7

CONCLUSIONS

Chapters 4, 5, and 6 have explored the major topics of this thesis with nine propositions in a testable format. These propositions are closely connected to the three main puzzles drawn from real-world politics and theoretical inquisitions. All of these are derived from the big motivation of this study — understanding the dynamics of war and trade in world politics. Computer simulations have been used for modeling the typical artificial world politics with and without the trade module, for running major rules of international politics, and for generating empirical data out of those virtual settings. Many probable conditions of world politics were implemented using different settings built from theoretical foundations in order to find meaningful patterns in state interactions. In particular, the comparison between the security-only world and the security-and-trade world, the impact of dyadic factors (economic interdependence and power ratio) on the probability of war, and the impact of structural variables (polarity and power concentration among major powers) on the frequency of war are explored in this thesis. I will summarize the findings here, with some comments on their theoretical implications, limitations and future research agenda.
7.1 Propositions Revisited

This study tested nine propositions related to the three main puzzles presented at
the first part of the thesis. The propositions were divided into three groups, according to
the basic theme of each group. The first group deals with the impact of the change in state
agents, represented by the introduction of the trade option to the existing *Realpolitik*
model. These propositions test any difference between the security-only world and the
security-and-trade world in three areas – system endurance, balance-of-power, and the
state survival ratio. The first three propositions follow the original research design of
Cusack and Stoll (1990) in order to compare the two models of world politics – the
*Realpolitik* model and the combined model with the addition of the liberal factor. The
simulation model developed for these propositions was used again, as the base model, in
testing other propositions about the impact of dyadic and structural factors.

The second group tested the role of dyadic properties in state interactions,
especially the level of economic interdependence and the heterogeneity in state powers
between states on the dyad. The main focus of these propositions was on testing the
findings of existing empirical studies – the negative relationship between economic
interdependence and the probability of war – and identifying the conditions of those
relationships. The third group intended to test the influence of structural factors – system
polarity and the power concentration among major powers – on the probability of war
upon the base model. Table 7.1 is the summary of the findings regarding each of the nine
propositions.
<table>
<thead>
<tr>
<th>Propositions</th>
<th>Findings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposition 1: With the introduction of trade, the chance of the system of</td>
<td>The system endures longer with the trade option.</td>
</tr>
<tr>
<td>states to endure rather than collapse into universal empire will be higher</td>
<td>Deterrent balance-of-power happens more frequently with the introduction of the trade option,</td>
</tr>
<tr>
<td>than in a world where only security relations (including alliance formation)</td>
<td>but defensive balance-of-power does not show a big difference.</td>
</tr>
<tr>
<td>are possible.</td>
<td>The trade option increases the survival chance of states. With the trade option, state-level</td>
</tr>
<tr>
<td></td>
<td>variables show bigger difference in the short term, while system-level variables do so in the</td>
</tr>
<tr>
<td></td>
<td>longer term.</td>
</tr>
<tr>
<td>Proposition 2: With the introduction of trade, the probability of balance-</td>
<td>Economic interdependence have negative impact on the probability of war.</td>
</tr>
<tr>
<td>of-power will be higher than that of the security-only world.</td>
<td>The probability of war is higher between minor powers than between major powers.</td>
</tr>
<tr>
<td>Proposition 3: With the introduction of trade, the survival chance of states</td>
<td>The probability of war is higher in the homogeneous dyad (in terms of power) than in the</td>
</tr>
<tr>
<td>within systems will be higher than that of the security-only world.</td>
<td>heterogeneous dyad.</td>
</tr>
<tr>
<td>Proposition 4: The more economically interdependent states are, the lower</td>
<td>As the power difference between states gets bigger, the probability of war decreases.</td>
</tr>
<tr>
<td>the probability of war between them becomes.</td>
<td>As the level of system polarity increases, war happens less frequently.</td>
</tr>
<tr>
<td>Proposition 5: The probability of war between major states is lower than</td>
<td>The probability of war follows an inverted U-shaped curve as the level of power concentration</td>
</tr>
<tr>
<td>that between minor states.</td>
<td>increases among major states.</td>
</tr>
<tr>
<td>Proposition 6: The probability of war in heterogeneous dyads (in terms of</td>
<td></td>
</tr>
<tr>
<td>state power) is higher than that in homogeneous dyads.</td>
<td></td>
</tr>
<tr>
<td>Proposition 7: The bigger the difference in power between two states (on a</td>
<td></td>
</tr>
<tr>
<td>continuous scale), the higher the probability of war between them.</td>
<td></td>
</tr>
<tr>
<td>Proposition 8: The probability of war declines as the number of poles</td>
<td></td>
</tr>
<tr>
<td>decreases.</td>
<td></td>
</tr>
<tr>
<td>Proposition 9: The probability of war follows an inverted U-shaped curve as</td>
<td></td>
</tr>
<tr>
<td>the level of power concentration increases among major states.</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.1: Propositions and the Summary of Findings
7.2 Theoretical Implications

7.2.1 State and Choice

This study started from the most fundamental assumption that state agents in world politics are rational decision-makers. This means that we regard state behavior as a choice based upon the calculation of expected utility of each path. States are assumed to choose the option that is believed to produce the highest expected utility, whether it is war or trade. Although there have been studies that establish other types of state agents - such as primitive power seekers or balancer-of-powers - the assumption of rationality seems not to be replaced by alternatives for the time being. Upon the running of many simulations and the analysis of their results, we can see that the assumption of rational actor works well in the formal mechanism of world politics. Of course, considering the fair critique on the concept of rationality, a certain amount of error is introduced in the modeling of the calculation process, too.50

The framework of rational choice in the simulation study of international relations was originally initiated by Bremer and Mihalka (1977) and Cusack and Stoll (1990) and replicated in this study whether the mechanism of state policy-making produces similarly plausible results as those of the real world. With some consideration of the possibility of error in estimating state's power, the rational choice framework used in this study worked fine in the fields of both conflict and cooperation in world politics. Although this study is not for the rational choice theory per se, its application in the working mechanism of state

50 For more on the theoretical basis for this introduction of the error factor in the rational choice paradigm, see Simon (1985) and Cusack and Stoll (1990, 30-31).
policy-making in the artificial world made sense. Although the processes of the war and trade mechanisms in this study are not complex enough to reproduce the real-world on the computer screen, it could generate many essential phenomena of world politics such as war, trade, alliance, balance-of-power, and polarity.

Many empirical studies care about the relationship between the causes and the effects of the target of study but the process between them does not matter. This consideration matters when we discuss about the concept of causality. The tradition of empirical study focuses on the question of “whether there is any correlation or causal relationship between the causes and the effects?” Sometimes this needs to be complemented by more theoretical works about why and how the target phenomena happen. The rational choice theory and the method of simulation contribute to this demand. Actually, the tradition of formal theory investigates the internal mechanism by asking “how does the phenomenon happen?” The rational choice theory shows it with axioms, deductive propositions and equations. The simulations used in this study implemented the “how” mechanisms on the computer screen with computer programs. Therefore the formal methods of rational choice theory and simulations focus more on the detailed processes than the simple correlations or causal relationships among factors. In this sense, the idea of state choice, borrowed from the rational choice mechanism, should be a good starting point in building theories of international relations.

This study tried to extend the applicability of the rational choice framework used in Bremer and Mihalka, and Cusack and Stoll, by incorporating these points. The previous studies succeeded in modeling the conflict dimension of world politics, and this

---

51 For the details of the concept of causality, see King, Keohane and Verba (1994, 76-91).
study in modeling the different types of international relations, conflict and trade. In this sense, this study can be regarded as an extension of the rational choice paradigm to the area of international trade and cooperation. Of course, this study does not take the rational choice theory as the basic research paradigm; the rational choice theory is applied only to the theoretical foundations of state agents in determining their behavior of war, alliance, or trade. At least at the micro-level mechanism, therefore, the application of the rational choice theory to the behavior of state choice in this thesis has contributed to the theorizing of war, conflict, and trade.

7.2.2 Interdependence, Heterogeneity and War

The findings in this study, regarding economic interdependence, support existing empirical studies by showing that the increase of economic interdependence between two states in a dyad tends to decrease their probability of war. Although there is no surprise in this finding, the result of simulations with regard to the impact of the power ratio (i.e., the different types of dyad in terms of state power) shows an interesting pattern. The probability of war is highest in the dyad composed of minor states, while it is much lower in the dyad between two major states and in the dyad initiated by a major state to a minor state. Putting aside the case of minor-major dyads, in which a minor state could not initiate any conflict against her bigger partner due to the low likelihood of victory, the findings seem to confirm Waltz's point more than Rosecrance' in explaining the post-War world dynamics and conflict patterns. That is, Waltz's focus was on the independent relationship between major powers as the main cause of the post-War peace, while
Rosecrance emphasized the increasing interdependence among small countries. At least, in this artificial world, small countries (i.e., minor states) fight more frequently with each other than major states. Of course, Waltz’s theory and Rosecrance’ theory need not be contradictory to each other, but the result of this study puts more weight on the logic of strong states following the Waltzian theory of structure.

As mentioned before, Chapter 5 of this thesis intended to fill the gap between the lack of theoretical bases and the problems in empirical studies by way of the simulation method. Based upon the key finding that economic interdependence tends to increase the probability of war in the world, as in the existing literature, this study has tried to test the possibility of difference between various dyadic situations. The simulation results showed that the heterogeneity in state size in the interacting dyads tend to influence the probability of war between themselves. Also it tends to make indirect effects on the relationship between economic interdependence and the probability of war. This result urges us to advance existing empirical works in depth according to different types of dyad, which have not yet been conducted much. If the behavioral patterns are different between major states and small states, then we have to develop a more comprehensive and more sophisticated theory on the relationship between economic interdependence and the peace of world politics.

Since many empirical studies on the topic of interdependence and war have relied on the dyadic data, the findings of this study can be a guideline for the research agenda for future. Although the analysis of this study was based upon the artificial dataset generated by a model of virtual world politics constructed only by core elements, these findings can contribute to the development of world politics models. In particular, it
showed that any generalization in the relationship between economic interdependence and peace in the world politics may have problems if we rely only on the dyadic dataset without considering the difference in the type of those dyads. This might offer some promising areas for future research, such as the size difference and the power ratio in dyads. This study has tried to suggest any possibility for different results in this context and found some noticeable results for empiricists who have not concentrated much on this point.

7.2.3 Neorealism, Constructivism and Structural Factors

The finding that structural factors are important in explaining war/trade patterns pushes us to compromise between the two theoretical frameworks – the top-down and the bottom-up ones. Above all, despite the fundamental bottom-up methodology of this study, it may be necessary to clarify how structural factors feedback to individual states. Another way to compromise is a new epistemology that regards individual-level factors as the source of structural factors which influence them again. This logic intends to resolve many contradictions between the macro-level factors and the micro-level factors in our explanation of world politics. Actually, as the third mainstream framework in international relations, the constructivist approach has argued for this resolution by assuming that agent and structure are mutually co-determined and "intersubjectively" interacting with each other (See Wendt, 1987, 1992 and 1994).

In this constructivist approach, the structure is not given anymore but is always changing itself according to its relation to individual agents. Individual agents also are
influenced by the structural factors. Using the term “intersubjective” relationship, the constructivist approach understands that both the actors and the structures are variables and they depend on historical and social “contexts” (Hopf, 1998, 176). This endogenous approach has some advantages in understanding the essential mechanisms of world politics, but it may have trouble in explaining causal relationship between variables due to circular logic. As Hopf puts it, the constructivist approach is troubled by the problem of “underspecification” as a theory of process since it does not specify the existence of its main elements – actors and structures (Ibid., 197). In this sense, the relativistic position of the constructivism is neither a “top-down” nor a “bottom-up” approach.52

While the constructivist approach cannot provide any complete answer for the problem of agent-structure relations, the simulation approach implemented in this thesis can offer a clue for this matter. The fundamental approach of this study, especially in its simulation modeling and implementation, is based on the CAS theory and the agent-based modeling (ABM). The overall topics analyzed in this study reflect these frameworks well, even though they are not explicitly expressed in each section. The ideas of CAS and ABM have been used in the practical programming stage so that they worked as the basic mechanism in the background of this study. While minimizing the use of global level parameters, the simulation runs in this thesis generated many system-level patterns, such as balance-of-power and polarity.

This successful implementation of CAS and ABM in this study for generating the macro-level phenomena – such as polarity and the emergence of empire – was based on

52 It is relativistic because the elements – actors and structures – are theoretically determined vis-à-vis each other, while they do not exist a priori in this framework. As the constructivist approach does not assume any “given” existence of main elements, it does not have any fixed point for analysis.
the setting of micro-level rules. In this context, it was necessary for us to develop the cellular automata framework further, which was originally initiated by Bremer and Mihalka (1977), Cusack and Stoll (1990) and others. With the CAS theory, we have a very strong base for the expansion of the bottom-up paradigm in the social sciences, not to mention the field of international relations. Particularly, the generation of macro-level data, mostly out of the theoretical mechanisms that work upon the unit-level or the dyad-level rules and properties, seems to be the core of applications that may be probable in this kind of simulation works.

The structural theory of neorealism has founded itself on the microeconomic theory in exploring the mechanism of individual actors’ behavior. At the same time, it has focused on the role of structural factors in determining individual states’ choice. The distribution of power and the polarity in world politics thus have been emphasized as the major factors in explaining world politics, while they are understood as emerging out of the micro-level factors. On the other hand, the constructivist approach has focused on the structural factors in a different perspective, as discussed above. This study has tried to review the problem of agent-structure relationship by testing the role of structural factors – polarity and the power concentration – upon the debate between these two camps. In particular, this thesis experimented with the structural neorealist proposition within the bottom-up framework. This can be a way of compromise between the contrasting epistemology between the previous two mainstreams in explaining the agent-structure relationship. This bottom-up simulation does not assume the structural factors “given” like the neorealist approach, nor does it fall into the trap of “underspecification” like the constructivist approach. Nevertheless, this study could test the neorealist theme about the
impact of the structural factors. By doing it this way, the epistemological and methodological features of this study may contribute to the integration of essential points in the debate of international relations topics.

7.2.4 Cellular Automata Simulations in International Relations

The addition of the trade option makes sense in modeling world politics and produces reasonable patterns to observe as the real-world phenomena. Upon the experiments so far, I have treated war and trade as “options” or “choices” for state agents. This simple scheme makes a difference vis-à-vis the conventional realist simulations – such as Bremer and Mihalka (1977) and Cusack and Stoll (1990). In these previous simulation works, a state is given chances to decide a war (or no action), to build an alliance, and to wage a real war. On the other hand, this thesis focuses on the additional option of trade for a state to choose, so that a state agent is assumed to choose between war, trade, and no action toward its neighbor countries. In other words, I tried to add the “liberal” element (the trade option as representing the cooperative option) to the conventional Realpolitik model, upon the assumption that the change of the choice set has an impact on the macro-level patterns of world politics. This effort, in this sense, can be regarded as an experiment with Rosecrance’ idea of the “trading world” to contrast with the classical “military-political-territorial world” proposed by many realists.

As expected, the introduction of the trade option at the individual state level has led a more peaceful world in simulation runs. Although I could not compare the difference between the security-only world and the security-and-trade world in some
parts due to the intrinsic assumptions of the experimental setting, the probability of war in the latter world has decreased and the possibility of system multiplicity increased with the trade option. In particular, the impact of trade has increased the length of system endurance; the systems lasted longer even though they have fallen to the overwhelming power of empires than in the security-only-world. In terms of balance-of-power, the introduction of the trade option has increased the possibility of deterrent balance-of-power even though there has been little difference in the patterns of defensive balance-of-power. Regarding state survival rate, trade makes it rise especially at the initial stage of state interactions. The results of these basic comparative runs between the security-only world and the security-and-trade world tell that the differences between them are significant and that we should not ignore the cooperative element such as trade in our modeling of world politics.

As such, this study has expanded the previous simulation works in international relations, especially those within the cellular automata framework. Considering that those works have focused only on the conflict dimension, I have conducted a combined algorithm of conflict and trade so that I could model world politics by adding the liberal factor of cooperation, or the trade factor. As the past history of computer simulation has shown, the scope of simulation study has been very limited in its application. Since the failure of the global modeling works, the direction of computer simulation has diverted toward a heuristic purpose and it seems to have succeeded at least in this direction. This study should stand on this line of heuristic tools in studying international relations, and it can be regarded as one of many possible extensions of the computer simulations in the study of world politics.
My simulation work follows the tradition of Bremer and Mihalka (1977) and Cusack and Stoll (1990) in its fundamental research framework and assumptions. Many variables in their original research design are replicated and the geographic map of 98 cells was re-used in my simulation. On the other hand, my study does not directly stand for the later works such as Duffy (1992) and Cederman (1997). As discussed before, Duffy contributed to the application of the parallel programming, but his idea is almost impossible to implement with the current level of machines. In particular, the hardware availability and programming method for parallel modeling seem to be the most important barriers that should be overcome for the development of the better simulation models with computer.

Cederman's works are special in the context of the computer simulation tradition in international relations. A noteworthy feature is his research framework based upon the combination of the complex systems theory and the constructivist theory. Regarding his connection to the complex systems theory, he distinguished himself from the previous simulation works in some points. Regarding his connection to the constructivist theory, he focused on the relationship between structure and agent in international relations. However, there still are some points to discuss in his research design from the perspective of the conventional simulations. The first difference, according to Cederman, between the old simulation and his new one is the execution style of simulations. He argues that the old simulations are based on the sequential programming, while his is a "truly parallel implementation" (1997, 82, footnote 5). This is intended to make all the states in the system to act simultaneously in every period. Although it is an ideal goal for every simulation, Cederman's statement is not true because his programming was not based
upon a parallel computing machine. Many programming skills have been developed, such as the multi-threading method in which a command is sliced into several pieces (or threads) before they are assigned to the execution sequence with the command pieces of other commands at the same time, for the parallel implementation. However, as Duffy already showed, a true parallel programming cannot be implemented by a single-processor machine. This is one reason that I did not follow Cederman’s idea of parallel programming. In addition, Cederman seems not understand well the previous simulation works. For example, he criticized the “predetermined game-tree” of the previous works (1997, 93). However, in Cusack and Stoll’s simulation, there are many stages of random distribution of parameters on the properties of agents.

Despite these problems, Cederman’s work has shown many possibilities for the development of the computational modeling in international relations. He was the first to identify his work as the application of the complex adaptive systems (CAS) theory to the field of international relations. Particularly, he introduced the agent-based modeling and many other promising methods. He was the first who identified his work as the computational modeling of the constructivist theory. He divided the state agent into sub-national actors such as nation and tried to show the emergence of state bottom-up rather than assuming the state agent as given. States in his Emergent Polarity Model (EPM) treats state agents as “history-dependent” upon the assumption that structure and agents co-determine (1997, 8).

53 Cederman used the “time-sharing” method for his simulation in which state agents act at the same time. His basic idea is that, in one time period, several agents act but the implementation order of those acts (actually the pieces of acts) are randomly determined in each period. This may kill the essence of the parallel programming – simultaneous execution of commands – by an arbitrary assignment of the order of the simulation commands.
Nevertheless, his simulation seems to have gone too long without any meaningful connection to the previous simulation works. It made Cederman’s work separate from the conventional simulations such as Bremen and Mihalka, and Cusack and Stoll. My simulation, considering this gap, tried not to deviate from the assumption of the rational choice actor paradigm in designing the nature of state agents. Also, I tried to apply the “bottom-up” idea of computer simulations by starting from the agent-level. As such, I tried to show that the model composed only of the neorealist and the liberal elements can produce the emergent phenomena even without the constructivist idea.

7.3 Limitations of the Study

This study is composed only of computer simulations of which algorithms are founded on many existing theories, models, and empirical data. The biggest problem of this kind of study is the validity of its model, so that I tried to match the mechanisms and parameters in this study to well-established existing theories. Whenever necessary, I had to search the range of values to make simulation parameters reflect the real-world. However, this job could not be perfect and is open to critiques at many points. In particular, without any close connection to existing theories and empirical data, it would be very hard for this kind of simulation work to have any meaningful implications in academic circles as well as in real politics. Nevertheless, this study focuses on its contribution to the theoretical dimension rather than to the policy dimension. Exploration in the area in which we may not have enough data and well-prepared theories is the foremost raison d’être of this simulation study.

217
Regarding the simulation technique, there are several limitations in this study that should not be ignored. The most important one is the "serial" implementation of simulation, which means that only one action is conducted at one time, no matter how short the basic computing time frame is. The original cellular automata framework was based on the "parallel" mechanism, like the John Conway's "Game of Life" in which one movement triggers every possible neighbor's response at the same time (Poundstone, 1985, 24). One of the reasons for the limitation in implementing parallel simulations is practical – we have only one CPU per computer machine. The real parallel simulation is possible only with the parallel machine – such as CM-5 – but it has not been always available to public.\(^4\) Duffy (1992 and 1993) tried the true parallel programming technique for his simulations, but he could not continue it for the accessibility to a parallel machine. For this reason, most simulation works in these days are "serial" despite a lot of techniques that emulate parallel programming.\(^5\)

Another missing point in this study is that there is no history in the working of the system. That is, a state wagers a war, builds an alliance, or trades resources with others, but she does not remember her past history of these actions, even though the current

\(^4\) W. Daniel Hillis, one of the pioneers of parallel computing, argued that the separation of CPU and memory in computer is inefficient relative to human brain, so that he suggested the parallel connection of chips. CM-5 was one of the first generation of parallel machines based on his computing philosophy (Hillis, 1985). Unfortunately, due to high cost, parallel machines have not been successful in the market.

\(^5\) The technique of "multi-threading" in Java may be one solution to the problem of serial computing. However, this is not a real parallel computing because it cuts the command threads from several sources and assigns the order of command pieces according to their priorities. That is, it starts to process several commands without finishing previous ones if priorities of those do change. Nevertheless, the CPU processes only one command piece at one processing time (Hz), so that we cannot call this a true emulation of parallel computing like the human brain. For a true emulation, we need several CPUs as many as needed for the simultaneous processing of commands. Another trend in computer science is the introduction of distributed computing, which splits the stream of commands into many pieces and send them to several computers connected in parallel. This method seems to be a promising solution in avoiding the high cost of parallel machine while assuring the real parallel model of processing.
system may influence on the actor's behavior in the next period. The state does not reflect its past history when it makes a decision. This does not make sense in the real-world politics so that we might have to consider revising the model to incorporate this aspect. Assuming that many wars or trades happen between previous enemies or between previous alliances, history seems the first priority for the additional candidates for a better model. Missing history in this study was a legacy from old models, particularly a must if we have to compare the mechanisms side-by-side. No history had been inserted in the cellular automata model before. Besides, compactness was the main concern of this study in extending those old models. Therefore, we may have to accept the missing "history" as a trade-off between the verisimilitude of the model and "Occam's Razor" for the beauty of the model.

7.4 Research Agenda for Future

The agenda for future research directions are the mirror images of the limitations of this study. In terms of the method used in this study, two of the most promising jobs for this kind of computer simulation study may be "parallel programming" and "evolutionary dynamics." The first was discussed in the previous section, but the cellular automata simulation has its core in the parallel effects of an action. Up to now, some simulation works have emulated parallel programming or used tricks for that. Many others, including this study, stay around the "serial" technique while acknowledging the advantages of parallel programming. One assumption of this position is that no actions happen at the same time, and thus they can be modeled and implemented in a sliced time.
frame in a serial manner. The idea of emulating parallel programming tries another way:
it slices an action into several pieces (from the start to the end) and assigns each of them
into the (still) serial machine according to the priority of each piece. Of course, the pieces
of other actions can be inserted into the processing so that several actions coexist in a
specific time frame. However, as mentioned before, this method is not truly parallel,
because the processor treats only one piece of action while holding other pieces before
they are finished (See footnote 37). We may expect a better technique for parallel
programming, such as distributed programming which has become popular these days, to
be available with single-processor machines in the social sciences in the near future.

The second aspect of “evolution” has a close relationship with the idea of history
discussed before. In fact, many scholars of the CAS-based simulations try to implement
the evolutionary dynamics of the system in which they are interested. Starting from a
random configuration, they would like to make the system evolve to a condition in which
they are interested. Any macro-level phenomenon “emerges” from the evolutionary
processes (of the interaction of agents), so that the core of the simulation can be put to the
establishment of the evolutionary dynamics (Holland, 1995 and 1998). The evolutionary
dynamics must be implemented by incorporating the “history” or “learning” of individual
agents. In the case of international relations, we might assume that state actors learn and
adjust to the international rules that have been developed out of their interactions
throughout history. As such, the dynamics between individuals and the system can be
modeled in a comprehensive manner. Agents “adapt” to their environment, and even
though their adaptation is local in scope, the evolution of the system may be possible
through these adaptations at the individual level.

220
Finally, the application of the CAS framework to many international relations topics seems promising. This study has taken just one step toward this, as it understands the dynamics of war and trade as a macro-level phenomenon built upon individual-level and dyad-level properties and rules (of course, including other systemic factors). Whether we are conducting empirical, formal, or simulation studies in international relations, we would get a lot of help from the CAS theory without throwing away our previous tools and frameworks. This is possible because the CAS framework is a comprehensive one rather than an alternative to existing theories. In particular, the introduction of agent-based modeling and the bottom-up approach reflects many new trends from other sciences. In addition, the technique of computer simulation has become faster, more economical, and more improved for our test of theoretical hypotheses and propositions. As such, we can use the CAS theory and the simulation methods as complementary tools for understanding many essential themes in international relations. This study is just a starting point for this development.

In terms of the subject in international relations, this study can be extended to incorporate many other topics in the study of international relations. This study compared the Realpolitik world with the newly generated security-and-trade world in order to show the importance of the liberal factors. Also this study tested the themes regarding the relationship between economic interdependence and the probability of war, the conditions under which heterogeneous dyads have different impact on the probability of war, and the relationship between structural factors and the pattern of conflict. Nevertheless, there still are many issues that are worth experimenting with computer simulations – different types of decision-makers, relative gains maximization,
opportunity cost of trade, and the relationship between foreign policy and domestic politics. The computer simulation on these topics may contribute to the development of international relations theories with the generation of artificial dataset and the clarification of theoretical points, in addition to empirical data analysis and formal modeling. In particular, the capability of computer simulations in generating huge datasets under controlled situations should not be underestimated. Although the computer simulation method does not directly treat the real-world data, it can supplement existing theories international relations theories in many ways.
REFERENCES


225

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.


228


APPENDIX 1

RELATIVE POWER RATIO OF MAJOR STATES, 1816-1965
(Source: COW)

<table>
<thead>
<tr>
<th>Period</th>
<th>Major Total</th>
<th>USA</th>
<th>UK</th>
<th>France</th>
<th>Germany</th>
<th>Austria-Hungary</th>
<th>Italy</th>
<th>Russia (USSR)</th>
<th>China</th>
<th>Japan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1816</td>
<td>83.0</td>
<td>-</td>
<td>34.8</td>
<td>12.4</td>
<td>5.3</td>
<td>10.5</td>
<td>-</td>
<td>20.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1817</td>
<td>83.0</td>
<td>-</td>
<td>22.9</td>
<td>18.0</td>
<td>11.9</td>
<td>11.0</td>
<td>-</td>
<td>19.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1818</td>
<td>82.5</td>
<td>-</td>
<td>23.1</td>
<td>17.5</td>
<td>12.3</td>
<td>10.7</td>
<td>-</td>
<td>18.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1819</td>
<td>80.2</td>
<td>-</td>
<td>22.1</td>
<td>18.2</td>
<td>11.8</td>
<td>9.7</td>
<td>-</td>
<td>18.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1820</td>
<td>79.2</td>
<td>-</td>
<td>32.5</td>
<td>12.4</td>
<td>5.4</td>
<td>9.3</td>
<td>-</td>
<td>19.6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1821</td>
<td>79.0</td>
<td>-</td>
<td>20.0</td>
<td>19.5</td>
<td>9.6</td>
<td>9.8</td>
<td>-</td>
<td>20.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1822</td>
<td>83.5</td>
<td>-</td>
<td>21.2</td>
<td>21.2</td>
<td>10.0</td>
<td>10.2</td>
<td>-</td>
<td>21.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1823</td>
<td>80.6</td>
<td>-</td>
<td>31.7</td>
<td>15.3</td>
<td>5.1</td>
<td>9.3</td>
<td>-</td>
<td>19.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1824</td>
<td>84.4</td>
<td>-</td>
<td>22.6</td>
<td>21.3</td>
<td>10.0</td>
<td>9.7</td>
<td>-</td>
<td>20.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1825</td>
<td>81.5</td>
<td>-</td>
<td>33.3</td>
<td>14.0</td>
<td>5.0</td>
<td>9.0</td>
<td>-</td>
<td>20.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1826</td>
<td>82.4</td>
<td>-</td>
<td>23.7</td>
<td>20.9</td>
<td>10.0</td>
<td>9.6</td>
<td>-</td>
<td>18.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1827</td>
<td>82.8</td>
<td>-</td>
<td>23.8</td>
<td>20.7</td>
<td>9.7</td>
<td>9.3</td>
<td>-</td>
<td>19.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1828</td>
<td>82.1</td>
<td>-</td>
<td>23.2</td>
<td>21.6</td>
<td>9.5</td>
<td>9.3</td>
<td>-</td>
<td>18.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1829</td>
<td>81.9</td>
<td>-</td>
<td>22.6</td>
<td>20.7</td>
<td>9.5</td>
<td>9.3</td>
<td>-</td>
<td>19.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1830</td>
<td>75.9</td>
<td>-</td>
<td>29.9</td>
<td>14.3</td>
<td>4.8</td>
<td>8.0</td>
<td>-</td>
<td>19.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1831</td>
<td>74.6</td>
<td>-</td>
<td>20.7</td>
<td>19.6</td>
<td>7.4</td>
<td>8.9</td>
<td>-</td>
<td>17.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1832</td>
<td>74.8</td>
<td>-</td>
<td>21.1</td>
<td>18.7</td>
<td>7.5</td>
<td>8.7</td>
<td>-</td>
<td>18.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1833</td>
<td>74.3</td>
<td>-</td>
<td>20.7</td>
<td>18.4</td>
<td>7.6</td>
<td>8.6</td>
<td>-</td>
<td>19.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1834</td>
<td>75.4</td>
<td>-</td>
<td>21.4</td>
<td>18.7</td>
<td>7.7</td>
<td>8.9</td>
<td>-</td>
<td>18.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1835</td>
<td>75.9</td>
<td>-</td>
<td>30.4</td>
<td>13.9</td>
<td>4.9</td>
<td>8.6</td>
<td>-</td>
<td>18.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1836</td>
<td>74.3</td>
<td>-</td>
<td>22.4</td>
<td>18.4</td>
<td>7.3</td>
<td>8.6</td>
<td>-</td>
<td>17.6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1837</td>
<td>73.7</td>
<td>-</td>
<td>22.5</td>
<td>18.3</td>
<td>7.5</td>
<td>8.5</td>
<td>-</td>
<td>16.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1838</td>
<td>73.9</td>
<td>-</td>
<td>22.7</td>
<td>18.4</td>
<td>7.6</td>
<td>8.7</td>
<td>-</td>
<td>16.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1839</td>
<td>73.6</td>
<td>-</td>
<td>22.6</td>
<td>17.8</td>
<td>7.7</td>
<td>9.0</td>
<td>-</td>
<td>16.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1840</td>
<td>75.1</td>
<td>-</td>
<td>31.0</td>
<td>15.4</td>
<td>4.8</td>
<td>7.4</td>
<td>-</td>
<td>16.6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1841</td>
<td>75.5</td>
<td>-</td>
<td>23.3</td>
<td>20.2</td>
<td>7.4</td>
<td>8.2</td>
<td>-</td>
<td>16.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1842</td>
<td>75.3</td>
<td>-</td>
<td>21.9</td>
<td>20.2</td>
<td>7.5</td>
<td>8.6</td>
<td>-</td>
<td>17.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1843</td>
<td>73.9</td>
<td>-</td>
<td>21.6</td>
<td>19.1</td>
<td>7.1</td>
<td>8.8</td>
<td>-</td>
<td>17.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1844</td>
<td>74.4</td>
<td>-</td>
<td>23.9</td>
<td>18.3</td>
<td>7.2</td>
<td>8.3</td>
<td>-</td>
<td>16.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1845</td>
<td>74.7</td>
<td>-</td>
<td>30.5</td>
<td>14.7</td>
<td>4.9</td>
<td>7.6</td>
<td>-</td>
<td>17.0</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

(Continued)
<table>
<thead>
<tr>
<th>Period</th>
<th>Major Total</th>
<th>USA</th>
<th>UK</th>
<th>France</th>
<th>Germany</th>
<th>Austria-Hungary</th>
<th>Italy</th>
<th>Russia (USSR)</th>
<th>China</th>
<th>Japan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1846</td>
<td>69.9</td>
<td>-</td>
<td>22.8</td>
<td>17.5</td>
<td>6.6</td>
<td>7.5</td>
<td>-</td>
<td>15.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1847</td>
<td>71.5</td>
<td>-</td>
<td>22.5</td>
<td>18.3</td>
<td>6.6</td>
<td>7.6</td>
<td>-</td>
<td>16.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1848</td>
<td>69.0</td>
<td>-</td>
<td>22.9</td>
<td>16.3</td>
<td>6.4</td>
<td>8.2</td>
<td>-</td>
<td>14.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1849</td>
<td>71.0</td>
<td>-</td>
<td>22.9</td>
<td>16.5</td>
<td>6.3</td>
<td>9.7</td>
<td>-</td>
<td>15.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1850</td>
<td>73.3</td>
<td>-</td>
<td>29.9</td>
<td>12.9</td>
<td>4.5</td>
<td>8.7</td>
<td>-</td>
<td>17.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1851</td>
<td>70.2</td>
<td>-</td>
<td>23.8</td>
<td>15.7</td>
<td>6.5</td>
<td>9.2</td>
<td>-</td>
<td>15.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1852</td>
<td>70.0</td>
<td>-</td>
<td>23.2</td>
<td>15.6</td>
<td>6.6</td>
<td>9.0</td>
<td>-</td>
<td>15.6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1853</td>
<td>69.9</td>
<td>-</td>
<td>22.0</td>
<td>15.9</td>
<td>6.5</td>
<td>9.2</td>
<td>-</td>
<td>16.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1854</td>
<td>75.1</td>
<td>-</td>
<td>33.3</td>
<td>12.7</td>
<td>4.5</td>
<td>8.1</td>
<td>-</td>
<td>16.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1855</td>
<td>74.4</td>
<td>-</td>
<td>29.3</td>
<td>14.9</td>
<td>4.5</td>
<td>7.1</td>
<td>-</td>
<td>18.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1856</td>
<td>74.6</td>
<td>-</td>
<td>29.3</td>
<td>14.3</td>
<td>4.6</td>
<td>6.5</td>
<td>-</td>
<td>19.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1857</td>
<td>70.8</td>
<td>-</td>
<td>28.3</td>
<td>14.3</td>
<td>5.2</td>
<td>7.7</td>
<td>-</td>
<td>15.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1858</td>
<td>70.3</td>
<td>-</td>
<td>28.4</td>
<td>13.4</td>
<td>5.5</td>
<td>8.1</td>
<td>-</td>
<td>14.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1859</td>
<td>70.4</td>
<td>-</td>
<td>28.7</td>
<td>14.1</td>
<td>5.1</td>
<td>8.3</td>
<td>-</td>
<td>14.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1860</td>
<td>61.7</td>
<td>-</td>
<td>27.8</td>
<td>12.3</td>
<td>4.6</td>
<td>5.1</td>
<td>2.8</td>
<td>9.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1861</td>
<td>56.9</td>
<td>-</td>
<td>25.9</td>
<td>10.8</td>
<td>4.6</td>
<td>4.1</td>
<td>3.3</td>
<td>8.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1862</td>
<td>54.7</td>
<td>-</td>
<td>24.9</td>
<td>10.7</td>
<td>4.8</td>
<td>4.3</td>
<td>3.3</td>
<td>6.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1863</td>
<td>55.5</td>
<td>-</td>
<td>24.9</td>
<td>10.3</td>
<td>4.9</td>
<td>4.3</td>
<td>3.5</td>
<td>7.6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1864</td>
<td>54.9</td>
<td>-</td>
<td>24.3</td>
<td>9.8</td>
<td>5.0</td>
<td>4.2</td>
<td>3.5</td>
<td>8.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1865</td>
<td>58.3</td>
<td>-</td>
<td>25.6</td>
<td>11.1</td>
<td>5.6</td>
<td>4.7</td>
<td>3.4</td>
<td>7.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1866</td>
<td>60.7</td>
<td>-</td>
<td>24.8</td>
<td>11.0</td>
<td>5.9</td>
<td>6.1</td>
<td>5.0</td>
<td>8.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1867</td>
<td>59.7</td>
<td>-</td>
<td>25.2</td>
<td>11.6</td>
<td>6.6</td>
<td>4.6</td>
<td>4.0</td>
<td>7.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1868</td>
<td>61.1</td>
<td>-</td>
<td>25.1</td>
<td>11.5</td>
<td>8.0</td>
<td>4.7</td>
<td>3.7</td>
<td>8.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1869</td>
<td>61.3</td>
<td>-</td>
<td>24.6</td>
<td>11.4</td>
<td>8.3</td>
<td>4.8</td>
<td>3.4</td>
<td>8.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1870</td>
<td>63.4</td>
<td>-</td>
<td>23.9</td>
<td>12.9</td>
<td>10.7</td>
<td>4.6</td>
<td>3.1</td>
<td>8.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1871</td>
<td>66.0</td>
<td>-</td>
<td>24.0</td>
<td>13.9</td>
<td>12.0</td>
<td>4.5</td>
<td>3.5</td>
<td>8.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1872</td>
<td>62.0</td>
<td>-</td>
<td>23.8</td>
<td>10.4</td>
<td>9.8</td>
<td>4.8</td>
<td>3.4</td>
<td>9.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1873</td>
<td>61.7</td>
<td>-</td>
<td>23.3</td>
<td>10.6</td>
<td>10.0</td>
<td>4.9</td>
<td>3.3</td>
<td>9.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1874</td>
<td>61.7</td>
<td>-</td>
<td>23.1</td>
<td>10.5</td>
<td>10.2</td>
<td>4.8</td>
<td>3.3</td>
<td>9.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1875</td>
<td>62.5</td>
<td>-</td>
<td>23.4</td>
<td>10.5</td>
<td>10.4</td>
<td>4.9</td>
<td>3.3</td>
<td>10.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1876</td>
<td>61.4</td>
<td>-</td>
<td>23.4</td>
<td>10.2</td>
<td>9.9</td>
<td>4.5</td>
<td>3.1</td>
<td>10.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1877</td>
<td>63.9</td>
<td>-</td>
<td>23.6</td>
<td>10.6</td>
<td>10.0</td>
<td>4.3</td>
<td>3.0</td>
<td>13.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1878</td>
<td>64.1</td>
<td>-</td>
<td>22.9</td>
<td>10.1</td>
<td>10.1</td>
<td>4.7</td>
<td>3.1</td>
<td>13.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1879</td>
<td>61.8</td>
<td>-</td>
<td>22.3</td>
<td>10.4</td>
<td>10.6</td>
<td>5.0</td>
<td>3.1</td>
<td>10.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1880</td>
<td>61.2</td>
<td>-</td>
<td>21.6</td>
<td>10.8</td>
<td>10.6</td>
<td>4.5</td>
<td>3.2</td>
<td>10.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1881</td>
<td>60.9</td>
<td>-</td>
<td>21.3</td>
<td>11.0</td>
<td>10.6</td>
<td>4.5</td>
<td>3.2</td>
<td>10.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1882</td>
<td>60.2</td>
<td>-</td>
<td>20.7</td>
<td>11.1</td>
<td>10.5</td>
<td>4.7</td>
<td>3.2</td>
<td>9.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1883</td>
<td>60.0</td>
<td>-</td>
<td>20.6</td>
<td>11.0</td>
<td>10.9</td>
<td>4.7</td>
<td>3.5</td>
<td>9.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1884</td>
<td>59.6</td>
<td>-</td>
<td>20.4</td>
<td>10.4</td>
<td>11.2</td>
<td>4.8</td>
<td>3.7</td>
<td>9.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1885</td>
<td>60.6</td>
<td>-</td>
<td>21.2</td>
<td>10.1</td>
<td>11.4</td>
<td>4.8</td>
<td>3.7</td>
<td>9.5</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

(Continued)
<table>
<thead>
<tr>
<th>Period</th>
<th>Major Total</th>
<th>USA</th>
<th>UK</th>
<th>France</th>
<th>Germany</th>
<th>Austria-Hungary</th>
<th>Italy</th>
<th>Russia</th>
<th>China</th>
<th>Japan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1886</td>
<td>58.8</td>
<td>19.8</td>
<td>9.9</td>
<td>11.0</td>
<td>4.8</td>
<td>3.7</td>
<td>9.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1887</td>
<td>59.2</td>
<td>19.1</td>
<td>9.6</td>
<td>12.2</td>
<td>5.0</td>
<td>3.9</td>
<td>9.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1888</td>
<td>58.4</td>
<td>18.3</td>
<td>9.3</td>
<td>12.5</td>
<td>4.8</td>
<td>4.1</td>
<td>9.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1889</td>
<td>57.7</td>
<td>18.5</td>
<td>9.5</td>
<td>11.6</td>
<td>4.8</td>
<td>3.8</td>
<td>9.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1890</td>
<td>57.7</td>
<td>17.6</td>
<td>9.5</td>
<td>12.4</td>
<td>5.0</td>
<td>3.6</td>
<td>9.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1891</td>
<td>58.1</td>
<td>17.6</td>
<td>9.7</td>
<td>12.1</td>
<td>5.0</td>
<td>3.6</td>
<td>10.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1892</td>
<td>57.3</td>
<td>17.1</td>
<td>9.4</td>
<td>12.4</td>
<td>5.0</td>
<td>3.4</td>
<td>10.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1893</td>
<td>58.2</td>
<td>16.8</td>
<td>9.5</td>
<td>13.2</td>
<td>5.1</td>
<td>3.4</td>
<td>10.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1894</td>
<td>58.2</td>
<td>17.4</td>
<td>9.2</td>
<td>12.9</td>
<td>5.0</td>
<td>3.2</td>
<td>10.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1895</td>
<td>60.4</td>
<td>17.0</td>
<td>8.9</td>
<td>12.4</td>
<td>4.9</td>
<td>3.5</td>
<td>10.7</td>
<td></td>
<td>3.3</td>
<td></td>
</tr>
<tr>
<td>1896</td>
<td>60.5</td>
<td>17.2</td>
<td>9.1</td>
<td>12.8</td>
<td>5.0</td>
<td>3.2</td>
<td>10.9</td>
<td></td>
<td>2.4</td>
<td></td>
</tr>
<tr>
<td>1897</td>
<td>59.6</td>
<td>16.4</td>
<td>8.7</td>
<td>13.0</td>
<td>4.9</td>
<td>3.1</td>
<td>10.7</td>
<td></td>
<td>2.7</td>
<td></td>
</tr>
<tr>
<td>1898</td>
<td>57.0</td>
<td>15.5</td>
<td>8.2</td>
<td>12.3</td>
<td>4.8</td>
<td>3.1</td>
<td>10.3</td>
<td></td>
<td>2.8</td>
<td></td>
</tr>
<tr>
<td>1899</td>
<td>77.1</td>
<td>18.7</td>
<td>16.6</td>
<td>8.2</td>
<td>12.5</td>
<td>4.6</td>
<td>2.9</td>
<td>10.8</td>
<td>2.8</td>
<td></td>
</tr>
<tr>
<td>1900</td>
<td>78.0</td>
<td>18.9</td>
<td>17.6</td>
<td>7.5</td>
<td>13.0</td>
<td>4.5</td>
<td>2.8</td>
<td>10.9</td>
<td>2.8</td>
<td></td>
</tr>
<tr>
<td>1901</td>
<td>78.5</td>
<td>20.3</td>
<td>17.2</td>
<td>7.3</td>
<td>12.6</td>
<td>4.4</td>
<td>2.8</td>
<td>10.9</td>
<td>2.8</td>
<td></td>
</tr>
<tr>
<td>1902</td>
<td>78.0</td>
<td>20.6</td>
<td>16.0</td>
<td>7.2</td>
<td>13.0</td>
<td>4.4</td>
<td>2.8</td>
<td>11.1</td>
<td>2.8</td>
<td></td>
</tr>
<tr>
<td>1903</td>
<td>77.6</td>
<td>21.0</td>
<td>14.1</td>
<td>7.3</td>
<td>13.4</td>
<td>4.4</td>
<td>2.9</td>
<td>11.3</td>
<td>3.2</td>
<td></td>
</tr>
<tr>
<td>1904</td>
<td>78.1</td>
<td>20.4</td>
<td>13.3</td>
<td>7.1</td>
<td>13.3</td>
<td>4.4</td>
<td>2.8</td>
<td>11.3</td>
<td>5.4</td>
<td></td>
</tr>
<tr>
<td>1905</td>
<td>79.6</td>
<td>21.3</td>
<td>11.9</td>
<td>6.3</td>
<td>12.2</td>
<td>4.0</td>
<td>2.5</td>
<td>16.5</td>
<td>4.8</td>
<td></td>
</tr>
<tr>
<td>1906</td>
<td>77.8</td>
<td>21.8</td>
<td>12.2</td>
<td>7.0</td>
<td>13.1</td>
<td>4.3</td>
<td>2.8</td>
<td>12.5</td>
<td>4.2</td>
<td></td>
</tr>
<tr>
<td>1907</td>
<td>78.2</td>
<td>22.7</td>
<td>12.0</td>
<td>7.1</td>
<td>13.5</td>
<td>4.4</td>
<td>2.8</td>
<td>12.0</td>
<td>3.6</td>
<td></td>
</tr>
<tr>
<td>1908</td>
<td>77.6</td>
<td>20.2</td>
<td>12.0</td>
<td>7.3</td>
<td>14.6</td>
<td>4.8</td>
<td>3.0</td>
<td>12.1</td>
<td>3.7</td>
<td></td>
</tr>
<tr>
<td>1909</td>
<td>77.5</td>
<td>22.4</td>
<td>11.5</td>
<td>7.0</td>
<td>13.5</td>
<td>4.7</td>
<td>3.0</td>
<td>11.9</td>
<td>3.4</td>
<td></td>
</tr>
<tr>
<td>1910</td>
<td>77.6</td>
<td>22.6</td>
<td>11.5</td>
<td>7.1</td>
<td>13.6</td>
<td>4.5</td>
<td>3.1</td>
<td>11.9</td>
<td>3.5</td>
<td></td>
</tr>
<tr>
<td>1911</td>
<td>77.3</td>
<td>21.5</td>
<td>11.7</td>
<td>7.2</td>
<td>13.7</td>
<td>4.4</td>
<td>3.3</td>
<td>12.0</td>
<td>3.5</td>
<td></td>
</tr>
<tr>
<td>1912</td>
<td>79.3</td>
<td>22.3</td>
<td>11.6</td>
<td>7.3</td>
<td>13.7</td>
<td>4.6</td>
<td>3.9</td>
<td>12.5</td>
<td>3.5</td>
<td></td>
</tr>
<tr>
<td>1913</td>
<td>77.1</td>
<td>21.7</td>
<td>11.1</td>
<td>6.8</td>
<td>14.3</td>
<td>4.7</td>
<td>3.4</td>
<td>11.8</td>
<td>3.4</td>
<td></td>
</tr>
<tr>
<td>1914</td>
<td>81.0</td>
<td>20.5</td>
<td>13.5</td>
<td>7.8</td>
<td>15.8</td>
<td>6.6</td>
<td>2.6</td>
<td>11.0</td>
<td>3.1</td>
<td></td>
</tr>
<tr>
<td>1915</td>
<td>84.7</td>
<td>22.1</td>
<td>14.8</td>
<td>8.3</td>
<td>15.1</td>
<td>6.0</td>
<td>3.4</td>
<td>12.2</td>
<td>2.7</td>
<td></td>
</tr>
<tr>
<td>1916</td>
<td>86.8</td>
<td>22.8</td>
<td>15.2</td>
<td>8.9</td>
<td>15.1</td>
<td>4.3</td>
<td>3.5</td>
<td>14.3</td>
<td>2.7</td>
<td></td>
</tr>
<tr>
<td>1917</td>
<td>84.9</td>
<td>23.5</td>
<td>15.0</td>
<td>8.7</td>
<td>16.0</td>
<td>4.1</td>
<td>3.3</td>
<td>11.3</td>
<td>2.9</td>
<td></td>
</tr>
<tr>
<td>1918</td>
<td>79.9</td>
<td>28.7</td>
<td>14.7</td>
<td>9.2</td>
<td>17.5</td>
<td>3.3</td>
<td>3.5</td>
<td></td>
<td>3.0</td>
<td></td>
</tr>
<tr>
<td>1919</td>
<td>63.6</td>
<td>37.8</td>
<td>11.3</td>
<td>6.3</td>
<td></td>
<td>-</td>
<td>4.6</td>
<td></td>
<td>3.6</td>
<td></td>
</tr>
<tr>
<td>1920</td>
<td>55.0</td>
<td>28.5</td>
<td>12.8</td>
<td>5.9</td>
<td></td>
<td>-</td>
<td>3.7</td>
<td></td>
<td>4.1</td>
<td></td>
</tr>
<tr>
<td>1921</td>
<td>47.9</td>
<td>25.2</td>
<td>9.6</td>
<td>5.1</td>
<td></td>
<td>-</td>
<td>3.6</td>
<td></td>
<td>4.3</td>
<td></td>
</tr>
<tr>
<td>1922</td>
<td>62.2</td>
<td>25.1</td>
<td>9.2</td>
<td>5.8</td>
<td></td>
<td>-</td>
<td>3.5</td>
<td>14.5</td>
<td>4.2</td>
<td></td>
</tr>
<tr>
<td>1923</td>
<td>60.0</td>
<td>26.8</td>
<td>9.7</td>
<td>5.9</td>
<td></td>
<td>-</td>
<td>3.0</td>
<td>10.5</td>
<td>4.1</td>
<td></td>
</tr>
<tr>
<td>1924</td>
<td>58.0</td>
<td>24.9</td>
<td>10.5</td>
<td>6.2</td>
<td></td>
<td>-</td>
<td>3.6</td>
<td>8.4</td>
<td>4.3</td>
<td></td>
</tr>
<tr>
<td>1925</td>
<td>65.9</td>
<td>25.1</td>
<td>9.4</td>
<td>6.1</td>
<td>7.9</td>
<td>-</td>
<td>3.2</td>
<td>10.2</td>
<td>4.1</td>
<td></td>
</tr>
</tbody>
</table>

(Continued)
<table>
<thead>
<tr>
<th>Period</th>
<th>Major Total</th>
<th>USA</th>
<th>UK</th>
<th>France</th>
<th>Germany</th>
<th>Austria-Hungary</th>
<th>Italy</th>
<th>Russia (USSR)</th>
<th>China</th>
<th>Japan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1926</td>
<td>66.4</td>
<td>25.8</td>
<td>7.9</td>
<td>6.1</td>
<td>7.7</td>
<td>-</td>
<td>3.3</td>
<td>11.5</td>
<td>-</td>
<td>4.2</td>
</tr>
<tr>
<td>1927</td>
<td>66.6</td>
<td>23.4</td>
<td>8.9</td>
<td>6.3</td>
<td>8.3</td>
<td>-</td>
<td>3.6</td>
<td>11.9</td>
<td>-</td>
<td>4.2</td>
</tr>
<tr>
<td>1928</td>
<td>65.3</td>
<td>23.6</td>
<td>8.1</td>
<td>5.9</td>
<td>7.7</td>
<td>-</td>
<td>3.3</td>
<td>12.5</td>
<td>-</td>
<td>4.1</td>
</tr>
<tr>
<td>1929</td>
<td>65.4</td>
<td>23.6</td>
<td>7.9</td>
<td>5.6</td>
<td>7.6</td>
<td>-</td>
<td>3.3</td>
<td>13.4</td>
<td>-</td>
<td>4.1</td>
</tr>
<tr>
<td>1930</td>
<td>65.8</td>
<td>22.4</td>
<td>7.6</td>
<td>6.1</td>
<td>7.3</td>
<td>-</td>
<td>3.3</td>
<td>15.0</td>
<td>-</td>
<td>4.1</td>
</tr>
<tr>
<td>1931</td>
<td>65.7</td>
<td>21.1</td>
<td>7.6</td>
<td>6.5</td>
<td>7.0</td>
<td>-</td>
<td>3.4</td>
<td>15.9</td>
<td>-</td>
<td>4.3</td>
</tr>
<tr>
<td>1932</td>
<td>65.3</td>
<td>19.2</td>
<td>8.0</td>
<td>6.9</td>
<td>7.0</td>
<td>-</td>
<td>3.8</td>
<td>15.3</td>
<td>-</td>
<td>5.1</td>
</tr>
<tr>
<td>1933</td>
<td>66.7</td>
<td>20.0</td>
<td>7.7</td>
<td>6.4</td>
<td>7.8</td>
<td>-</td>
<td>3.8</td>
<td>15.9</td>
<td>-</td>
<td>5.1</td>
</tr>
<tr>
<td>1934</td>
<td>67.0</td>
<td>19.2</td>
<td>7.8</td>
<td>5.8</td>
<td>8.9</td>
<td>-</td>
<td>3.6</td>
<td>16.6</td>
<td>-</td>
<td>5.1</td>
</tr>
<tr>
<td>1935</td>
<td>69.6</td>
<td>18.9</td>
<td>7.2</td>
<td>5.2</td>
<td>10.2</td>
<td>-</td>
<td>5.2</td>
<td>18.1</td>
<td>-</td>
<td>4.9</td>
</tr>
<tr>
<td>1936</td>
<td>68.2</td>
<td>20.1</td>
<td>7.4</td>
<td>5.3</td>
<td>11.5</td>
<td>-</td>
<td>4.0</td>
<td>15.0</td>
<td>-</td>
<td>4.8</td>
</tr>
<tr>
<td>1937</td>
<td>68.1</td>
<td>19.4</td>
<td>7.5</td>
<td>4.9</td>
<td>11.7</td>
<td>-</td>
<td>4.0</td>
<td>14.9</td>
<td>-</td>
<td>5.6</td>
</tr>
<tr>
<td>1938</td>
<td>70.0</td>
<td>16.5</td>
<td>7.5</td>
<td>4.6</td>
<td>15.3</td>
<td>-</td>
<td>3.3</td>
<td>16.6</td>
<td>-</td>
<td>6.2</td>
</tr>
<tr>
<td>1939</td>
<td>73.1</td>
<td>18.0</td>
<td>9.9</td>
<td>4.0</td>
<td>18.1</td>
<td>-</td>
<td>2.8</td>
<td>13.9</td>
<td>-</td>
<td>6.5</td>
</tr>
<tr>
<td>1940</td>
<td>76.4</td>
<td>19.9</td>
<td>9.4</td>
<td>7.8</td>
<td>17.4</td>
<td>-</td>
<td>3.0</td>
<td>13.5</td>
<td>-</td>
<td>5.4</td>
</tr>
<tr>
<td>1941</td>
<td>76.6</td>
<td>23.9</td>
<td>9.8</td>
<td>-</td>
<td>20.9</td>
<td>-</td>
<td>2.1</td>
<td>12.6</td>
<td>-</td>
<td>7.3</td>
</tr>
<tr>
<td>1942</td>
<td>77.4</td>
<td>27.9</td>
<td>9.3</td>
<td>-</td>
<td>18.2</td>
<td>-</td>
<td>3.6</td>
<td>11.6</td>
<td>-</td>
<td>6.8</td>
</tr>
<tr>
<td>1943</td>
<td>77.9</td>
<td>33.6</td>
<td>8.6</td>
<td>-</td>
<td>16.2</td>
<td>-</td>
<td>2.4</td>
<td>10.6</td>
<td>-</td>
<td>6.6</td>
</tr>
<tr>
<td>1944</td>
<td>74.1</td>
<td>34.1</td>
<td>8.2</td>
<td>-</td>
<td>14.2</td>
<td>-</td>
<td>-</td>
<td>10.2</td>
<td>-</td>
<td>7.5</td>
</tr>
<tr>
<td>1945</td>
<td>75.5</td>
<td>38.0</td>
<td>8.9</td>
<td>1.8</td>
<td>8.4</td>
<td>-</td>
<td>-</td>
<td>12.3</td>
<td>-</td>
<td>6.1</td>
</tr>
<tr>
<td>1946</td>
<td>62.7</td>
<td>35.4</td>
<td>11.6</td>
<td>3.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>12.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1947</td>
<td>57.3</td>
<td>29.9</td>
<td>9.1</td>
<td>3.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>14.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1948</td>
<td>55.7</td>
<td>28.3</td>
<td>7.5</td>
<td>3.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>16.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1949</td>
<td>54.3</td>
<td>26.7</td>
<td>7.0</td>
<td>3.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>17.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1950</td>
<td>67.4</td>
<td>27.7</td>
<td>6.3</td>
<td>3.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>18.2</td>
<td>11.9</td>
<td>-</td>
</tr>
<tr>
<td>1951</td>
<td>68.4</td>
<td>31.1</td>
<td>6.0</td>
<td>3.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>17.4</td>
<td>10.5</td>
<td>-</td>
</tr>
<tr>
<td>1952</td>
<td>66.4</td>
<td>30.2</td>
<td>5.8</td>
<td>3.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>17.1</td>
<td>9.9</td>
<td>-</td>
</tr>
<tr>
<td>1953</td>
<td>66.5</td>
<td>30.2</td>
<td>5.6</td>
<td>3.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>17.4</td>
<td>10.2</td>
<td>-</td>
</tr>
<tr>
<td>1954</td>
<td>64.9</td>
<td>27.2</td>
<td>5.6</td>
<td>3.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>18.5</td>
<td>10.3</td>
<td>-</td>
</tr>
<tr>
<td>1955</td>
<td>62.0</td>
<td>25.9</td>
<td>5.1</td>
<td>3.1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>17.9</td>
<td>10.0</td>
<td>-</td>
</tr>
<tr>
<td>1956</td>
<td>61.4</td>
<td>25.4</td>
<td>4.9</td>
<td>3.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>17.0</td>
<td>10.8</td>
<td>-</td>
</tr>
<tr>
<td>1957</td>
<td>60.7</td>
<td>24.7</td>
<td>4.7</td>
<td>3.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>16.5</td>
<td>11.5</td>
<td>-</td>
</tr>
<tr>
<td>1958</td>
<td>60.1</td>
<td>22.8</td>
<td>4.5</td>
<td>3.4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>16.8</td>
<td>12.5</td>
<td>-</td>
</tr>
<tr>
<td>1959</td>
<td>60.1</td>
<td>22.4</td>
<td>4.3</td>
<td>3.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>16.9</td>
<td>13.2</td>
<td>-</td>
</tr>
<tr>
<td>1960</td>
<td>59.2</td>
<td>21.5</td>
<td>4.1</td>
<td>3.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>16.9</td>
<td>13.4</td>
<td>-</td>
</tr>
<tr>
<td>1961</td>
<td>57.5</td>
<td>21.2</td>
<td>4.0</td>
<td>3.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>17.2</td>
<td>12.1</td>
<td>-</td>
</tr>
<tr>
<td>1962</td>
<td>56.9</td>
<td>21.1</td>
<td>3.8</td>
<td>2.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>17.6</td>
<td>11.5</td>
<td>-</td>
</tr>
<tr>
<td>1963</td>
<td>56.2</td>
<td>21.0</td>
<td>3.7</td>
<td>2.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>17.1</td>
<td>11.5</td>
<td>-</td>
</tr>
<tr>
<td>1964</td>
<td>55.7</td>
<td>20.7</td>
<td>3.7</td>
<td>2.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>16.8</td>
<td>11.6</td>
<td>-</td>
</tr>
<tr>
<td>1965</td>
<td>54.9</td>
<td>20.4</td>
<td>3.7</td>
<td>2.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>16.5</td>
<td>11.6</td>
<td>-</td>
</tr>
</tbody>
</table>
APPENDIX 2

PROGRAM SOURCE CODE IN C++

The program used in this study is composed of many files, which were modified to reflect the topic in each chapter. The program source code for Chapter 4 as the base run is listed here, but the codes for other chapters follow almost the same format with some minimum changes for parameter setting. All source codes and compiled programs are available in the electronic format from the author upon request. Microsoft Visual C++ 6.0 was used as the main compiler, and one file for the program interface (windows.h) requires this compiler. Also this program needs some standard C++ library files, such as iostream.h, cmath.h, iomanip.h, fstream.h, string.h, and so on. In many cases, standard template library (STL) files such as vector.h, list.h, and algorithms.h were used, for which the code is not listed here. In the case of writing Mathfunc.h and Mathfunc.cpp, I used some functions which were already used in Cusack and Stoll's original C code. But all other source code was re-written in C++, even though I followed the same algorithm. The number of source files listed here is sixteen, but we need some more library files mentioned above to compile the whole program.
## Program Interface

<table>
<thead>
<tr>
<th>Header File</th>
<th>Definition File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Param.h (p. 237)</td>
<td>Aworld.cpp (p. 238)</td>
<td>Definition of important parameters</td>
</tr>
<tr>
<td>Cell.h (p. 262)</td>
<td>Cell.cpp (p. 264)</td>
<td>Main program file</td>
</tr>
<tr>
<td>Expected.h (p. 270)</td>
<td>Expected.cpp (p. 273)</td>
<td>Declarations and definitions about territorial cell’s properties (variables) and behaviors (functions)</td>
</tr>
<tr>
<td>Mathfunc.h (p. 291)</td>
<td>Mathfunc.cpp (p. 293)</td>
<td>Function definitions regarding the calculation of expected utility</td>
</tr>
<tr>
<td>Prewar.h (p. 296)</td>
<td>Prewar.cpp (p. 298)</td>
<td>Mathematical functions such as randomization, likelihood-of-victory related functions, etc.</td>
</tr>
<tr>
<td>Prints.h (p. 308)</td>
<td>Prints.cpp (p. 311)</td>
<td>Function declarations and definitions for war initiation and alliance formation</td>
</tr>
<tr>
<td>State.h (p. 325)</td>
<td>State.cpp (p. 328)</td>
<td>Functions related to printing parameters and outputs to files and the screen</td>
</tr>
<tr>
<td>Transfer.h (p. 338)</td>
<td>Transfer.cpp (p. 341)</td>
<td>Declarations and definitions about state’s properties and behaviors</td>
</tr>
</tbody>
</table>

## Source Code List

236
// Param.h
// Parameter settings
// Revised Apr. 17, 2000
#
// Minimum power, resources level

const double MINIMUM_POWER = 0.01;
const double MINIMUM_OR = 0.01;  // Minimum level of metabolism
const double MINIMUM_UNIT = 0.001;  // Minimum size of trade unit

// Total number of cells in the world
const int NCW = 98;

// Initial value of the number of states in the world
const int NSW = 98;

// Mathematical PI for the calculation of LV
// EPS is the digit that is wanted in the calculation of LV
// JMAX is set for the function qtrap() so that 2 to the power JMAX-1 is the maximum allowed number of steps.
const double PI = 3.14159;
const double EPS = 1.0e-5;
const int JMAX = 14;

// E (for the LV calculation)
const double E = 2.71818;

#endif // _PARAM_H_
// AWorld.cpp
// Version 0.6
// Main module of AWorld program
// Originally written May 03, 1999
// Revised May 20, 2001
// Revised July 25, 2001
// Revised Sep. 3, 2001
// 1. Change the power calculation level (RA * RB = POWER)
// 2. No "power" variables for each cell
// 3. Power variable is only for state
// 4. Delete state metabolism variables
// 5. Delete the "add_state_power()" and "subtract_state_power()"
// Revised Oct. 26, 2001
// 1. Revise the warcost calculation module (WCDISPAR)
// Revised Dec. 19, 2001
// 1. Three DV groups are identified (system, BOP, survival)
// 2. Trade ON/OFF switch provided
// 3. Some user interface modified (initial screen)
//*******************************************************************************

#include <iostream>
#include <iomanip>
#include <fstream>
#include <vector>
#include <list>
#include <string>
#include <algorithm>
#include <cmath>
#include <windows.h>
#include "Param.h"
#include "Cell.h"
#include "State.h"
#include "Prints.h"
#include "Prewar.h"
#include "Mathfunc.h"
#include "Transfer.h"
#include "Expected.h"

using namespace std;
const int run_max = 10000;
const int round_max = 1000;
int i; // index number

int main()
{
    // Basic file and parameter setting
    int trade_on = -1; // turn on (1) or off (0) trade module
    string filename0, filename1, filename2, filename3;
    char response;
    print_screen(filename0, filename1, filename2, filename3,
                 trade_on, response);

    return 0;
}
if(!(trade_on == 0) || (trade_on == 1))
{
    cerr << "The value for trade module turn is wrong." << endl;
    exit(1);
}
ofstream fout1(filename1.c_str(), ios::out);
if(!fout1)
{
    cerr << "Can't open " << filename1 << " file for output.\n";
    exit(1);
}
ofstream fout2(filename2.c_str(), ios::out);
if(!fout2)
{
    cerr << "Can't open " << filename2 << " file for output.\n";
    exit(1);
}
ofstream fout3(filename3.c_str(), ios::out);
if(!fout3)
{
    cerr << "Can't open " << filename3 << " file for output.\n";
    exit(1);
}

// Random seed generation
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
randomize();
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

// Loop for run
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
for(int run = 1; run <= run_max; run++)
{

// Configuration settings for runs
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
float POW_SD, ERROR_SD, WARCOSTMAX, REPAR, LVSIGMA; // Control factors
float WCDISPAR, GR_SD, META_SD; // Three experimental factors
POW_SD = random_float((float)0.1, (float)0.6);
ERROR_SD = random_float((float)0.1, (float)0.3);
WARCOSTMAX = random_float((float)0.05, (float)0.2);
REPAR = random_float((float)0.1, (float)0.3);
LVSIGMA = 1.0; // Fixed to 1.0 for rational choice actors
WCDISPAR = random_float((float)0.0, (float)0.5);
GR_SD = random_float((float)0.0, (float)0.005);
META_SD = random_float((float)0.1, (float)0.6);
float ura_growth_rate = nr((float)0.01, (float)GR_SD);
float urb_growth_rate = nr((float)0.01, (float)GR_SD);

// Run loop begins
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
vector<Cell> c;  // Cell vector (number fixed) declaration
vector<State> s; // State vector (number dynamic) declaration
vector<State>::iterator iter_state;
// Parameter initialization

int new_state_id = 97;
int new_state_number = 0;
int state_death = 0;
int no_war_freq = 0;
int trade_freq = 0;
int war_freq = 0;
int target_wins = 0;
int initiator_wins = 0;
float world_total_power = 0.0;
int final_state_number = 0;
int whether_multiplicity = 1; // if empire, it becomes 0
int num_iterations = 0; // when the run ends

// Print parameters

print_parameters(fout1, run, 1, POW_SD, ERROR_SD, LVSIGMA, WARCOSTMAX, WCDSPAR, REPAR, META_SD);

// Cell initialization module

for (i = 0; i < NCW; i++)
{
    float a = nr(1.0, POW_SD);
    if (a > MINIMUM_UR) c[i].set_ura(a);
    else c[i].set_ura(MINIMUM_UR);
}
for (i = 0; i < NCW; i++)
{
    float b = nr(1.0, POW_SD);
    if (b > MINIMUM_UR) c[i].set_urb(b);
    else c[i].set_urb(MINIMUM_UR);
}

// State initialization module

for (i = 0; i < NSW; i++)
{
    State state(i, i, c, s);
    s.push_back(state);
}
for (i = 0; i < NCW; i++) c[i].set_which_state(i);
for (i = 0; i < NSW; i++)
{
    float a = nr(1.0, POW_SD);
    if (a > MINIMUM_META) s[i].set_per_ma(a);
    else s[i].set_per_ma(MINIMUM_META);
}

240
float b = nr(1.0, POW_SD);
if (b > MINIMUM_META)
    s[i].set_per_mb(b);
else
    s[i].set_per_mb(MINIMUM_META);
}
for (i = 0; i < NSW; i++)
    s[i].cal_state_power(c);
for (i = 0; i < NSW; i++)
    s[i].set_snc(c);
for (i = 0; i < NSW; i++)
    s[i].set_sns(c);

// Round loop
for (int round = 1; round <= round_max; round++)
{
    // Vector creation
    vector<int> offensive_part; offensive_part.clear();
    vector<int> defensive_part; defensive_part.clear();
    // Initialization of round
    int warcells = 0; // The number of cells participating in war
    int number_of_states = 0; // The number of non-empty states
    for (i = 0; i < s.size(); i++)
    {
        iter_state = s.begin() + i;
        if ((*iter_state).get_sc_size() == 0)
        {
            s.erase(iter_state);
            break;
        }
    }
    for (i = 0; i < s.size(); i++)
    {
        s[i].set_id(i);  // Reset the id
    }
    for (i = 0; i < s.size(); i++)
    {
        s[i].set_snc(c);
        s[i].set_sns(c);
    }
    number_of_states = s.size();
    final_state_number = number_of_states;
    if (number_of_states == 1) break; // End of round if only one state
    // Normalize the volume of each cell's ura and urb (relative volumes)
    float world_total_ura = 0.0;
    float world_total_urb = 0.0;
float temp_ura, temp_urb;
for (i = 0; i < NCW; i++)
{
    world_total_ura += c[i].get_ura();
    world_total_urb += c[i].get_urb();
}
for (i = 0; i < NCW; i++)
{
    temp_ura = c[i].get_ura();
    temp_urb = c[i].get_urb();
    c[i].set_ura(temp_ura/world_total_ura);
    c[i].set_urb(temp_urb/world_total_urb);
}

// Redistribute ura and urb (set minimum = 0.01)
// Recalculate power, ura, urb of each state after normalization
world_total_power = 0.0;
for (i = 0; i < s.size(); i++)
{
    s[i].cal_state_power(c);
    world_total_power += s[i].get_state_power();
}
for (i = 0; i < s.size(); i++)
{
    s.at(i).set_snc(c);
    s.at(i).set_sns(c);
}
print_screen_l(cout, run, round, number_of_states);

// Power distribution and system polarity variables
vector<major_power> mp;
int system_polarity = -1;    // 0, 1, 2, 3
for (i = 0; i < s.size(); i++)
    if (s.at(i).get_state_power() > 0.03)
    {
        major_power temp;
        temp.id = s.at(i).get_id();
        temp.power = s.at(i).get_state_power();
        mp.push_back(temp);
    }
sort(mp.begin(), mp.end());
if (mp.size() > 8)
{
    do
    {
        mp.pop_back();
    } while (mp.size() > 8);
}
if (mp.size() == 0)
    system_polarity = 0;    // unclear
else
{
    float sum_major_powers = 0.0;
    for (int si = 0; si < mp.size(); si++)
        sum_major_powers += mp.at(si).power;
    if (mp.size() == 1)
    {
        if (mp.at(0).power >= 0.4)
            system_polarity = 1;  // unipolar
        else
            system_polarity = 0;  // unclear;
    }
    else if (mp.size() == 2)
    {
        if (mp.at(0).power >= 0.4)
            system_polarity = 1;  // unipolar
        else if ((mp.at(0).power < 0.4) &&
                  (mp.at(0).power + mp.at(1).power >= 0.4) &&
                  (mp.at(0).power - mp.at(1).power <= 0.1))
            system_polarity = 2;  // bipolar
        else
            system_polarity = 0;  // unclear
    }
    else  // if major_powers.size() >= 3
    {
        if (mp.at(0).power >= 0.4)
            system_polarity = 1;  // unipolar
        else if ((mp.at(0).power < 0.4) &&
                  (mp.at(0).power + mp.at(1).power >= 0.4) &&
                  (mp.at(0).power - mp.at(1).power <= 0.1))
            system_polarity = 2;  // bipolar
        else if (sum_major_powers > 0.7)
            system_polarity = 3;  // multipolar
        else
            system_polarity = 0;  // unclear
    }
}

// Assign the attributes of major power
// -----------------------------------------------------------------------------------------------
for (i = 0; i < s.size(); i++)
    s.at(i).set_major(0);
if (mp.size() > 0)
{
    for (int aa = 0; aa < mp.size(); aa++)
        s[mp.at(aa).id].set_major(1);
}

// Initialization of initiator-choosing process
// vseu is the vector of structure "state_eu" that holds the information
// about a state's maximum EU of trade and war, and targets
// -----------------------------------------------------------------------------------------------
vector<state_eu> vseu;

// Figure out max. utility of trade and/or war for each state
// Put the value of utility (0 or positive) into apus
// -----------------------------------------------------------------------------------------------
for (i = 0; i < s.size(); i++)
{

243
int ii; // Index number
float apower = s[i].get_state_power() * (1.0 + nr(0.0, ERROR_SD));
vector<neighbor_eu> vneu; // Neighbor state vector with EU
for (ii = 0; ii < s[i].get_sns_size(); ii++)
{
    neighbor_eu neu;
    neu.id = s[s[i].get_sns(ii)].get_id();
    neu.power = s[s[i].get_sns(ii)].get_state_power() * (1.0 + nr(0.0, ERROR_SD));
    float power_ratio = neu.power / apower;
    neu.eut = eu_trade_state(s, c, s[i].get_id(),
        s[s[i].get_sns(ii)].get_id(), ERROR_SD);
    neu.lv = likelihood_of_victory
        (apower/neu.power, LVSIGMA);
    neu.euw = exp_util_war_state
        (s, c, ERROR_SD, WARCOSTMAX, WCDISPAR,
        REPAR, LVSIGMA, s[i].get_id(),
        s[s[i].get_sns(ii)].get_id(), l, l,
        neu.lv, apower, neu.power);
    if (trade_on == 1)
    {
        if ((neu.euw > 0.0) && (neu.euw > neu.eut))
        {
            neu.wt = 2; // Choose war
            neu.bigger = neu.euw;
            neu.target = neu.id;
        }
        else if ((neu.eut > 0.0) && (neu.eut > neu.euw))
        {
            neu.wt = 1; // Choose trade
            neu.bigger = neu.eut;
            neu.target = neu.id;
        }
        else
        {
            neu.wt = 0; // No action
            neu.bigger = -1.0; // Negative EU
            neu.target = neu.id;
        }
    }
    else // if trade_on == 0
    {
        if (neu.euw > 0.0)
        {
            neu.wt = 2; // Choose war
            neu.bigger = neu.euw;
            neu.target = neu.id;
        }
        else
        {
            neu.wt = 0; // No action
            neu.bigger = -1.0; // Negative EU
            neu.target = neu.id;
        }
    }
    if (neu.bigger > 0.0) vneu.push_back(neu);
}
state_eu seu;
seu.max_util = 0.0; // Default assignment
if (vneu.size() > 0) // First element to seu
{
    seu.id = s[i].get_id();
    seu.real_power = s[i].get_state_power();
    seu.estimated_power = apower;
    seu.max_util = vneu.at(0).bigger;
    seu.target = vneu.at(0).target;
    seu.target_power = vneu.at(0).power;
    seu.wt = vneu.at(0).wt;
    seu.lv = vneu.at(0).lv;
}
// If more than one element, find the max. positive utility
if (vneu.size() > 1)
{
    for (ii = 1; ii < vneu.size(); ii++)
    {
        if (vneu.at(ii).bigger > seu.max_util)
        {
            seu.max_util = vneu.at(ii).bigger;
            seu.target = vneu.at(ii).target;
            seu.target_power = vneu.at(ii).power;
            seu.wt = vneu.at(ii).wt;
            seu.lv = vneu.at(ii).lv;
        }
    }
}
if (seu.max_util > 0.0) // If maximum EU > 0.0
    vseu.push_back(seu);

// Choose an initiator (based on relative size of positive EU)
// int pinitiator = -1; // initialization
int ptarget = -1; // initialization
int pinitiator_wot = -1; // 0 = no action, 1 = trade, 2 = war
// Put positive EU states into a new vector
vector<state_eu> positive_vseu;
if (vseu.size() > 0)
{
    for (i = 0; i < vseu.size(); i++)
        positive_vseu.push_back(vseu.at(i));
}

// End the loop if the vseu vector is empty
// else // if no state has positive EU
{ // if no state has positive EU
    print_screen_2(0, -1); // 0 no action, -1 no initiator
    no_war_freq++;
    // -----------------------------------------------------------------------------
    // Grow ura and urb, and then recalculate state power level
    // -----------------------------------------------------------------------------
    for (i = 0; i < c.size(); i++)
    {
        c[i].grow_ura(nr((float)0.01, (float)GR_SD));
        c[i].grow_urb(nr((float)0.01, (float)GR_SD));
    }
    for (i = 0; i < s.size(); i++)
        s[i].cal_state_power(c);
    // If no initiator is chosen here, then the two data output

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
functions (print_eu_polarity, print_real_wot) will not send any line to the “fout2” file pointer
continue; // No war; return to the round loop

Initiator selection module

Assign the value of relative_real_power
float real_power_total = 0.0;
for (i = 0; i < positive_vseu.size(); i++)
    real_power_total += positive_vseu.at(i).real_power;
for (i = 0; i < positive_vseu.size(); i++)
    positive_vseu.at(i).relative_real_power = positive_vseu.at(i).real_power / real_power_total;

Assign the value of weighted_eu_total
float weighted_eu_total = 0.0;
for (i = 0; i < positive_vseu.size(); i++)
    weighted_eu_total += positive_vseu.at(i).max_util * positive_vseu.at(i).relative_real_power;
for (i = 0; i < positive_vseu.size(); i++)
    positive_vseu.at(i).weighted_eu = positive_vseu.at(i).max_util / weighted_eu_total;

float random_number = random_float(0.0, 1.0);

choosing initiator based on relative power level
pinitiator = choose_initiator (positive_vseu, random_number, ptarget);
if (ptarget < 0)
{
    cerr << "Error in choosing target." << endl << endl;
    exit(1);
}
for (i = 0; i < positive_vseu.size(); i++)
{
    if (positive_vseu.at(i).id == s[pinitiator].get_id())
    {
        pinitiator_wot = positive_vseu.at(i).wt;
        break;
    }
}

// Print the intermediate variables (expected utility and dCON)

float i_power = s[pinitiator].get_state_power();
float t_power = s[ptarget].get_state_power();
float bigger = 0.0;
float dcon = calculate_dcon(i_power / (i_power + t_power),
    t_power / (i_power + t_power));
float p_ratio = i_power / t_power;
float i_eu = 0.0;
for (i = 0; i < positive_vseu.size(); i++)
{
    if (positive_vseu.at(i).id == s[pinitiator].get_id())
    {
        i_eu = positive_vseu.at(i).max_util;
        break;
    }
}
if (i_eu < 0.0)
cout << "Error in calculating trade expected utility." << endl;
exit(1);

float t_eu = 0.0;
t_eu = eu_trade_state(s, c, ptarget, pinitiator, ERROR_SD);

// Stronger and weaker state's EU/power ratios
float low_eu_ratio, high_eu_ratio;
if (i_eu/i_power > t_eu/t_power)
{
    low_eu_ratio = t_eu/t_power;
    high_eu_ratio = i_eu/i_power;
}
else
{
    low_eu_ratio = i_eu/i_power;
    high_eu_ratio = t_eu/t_power;
}

// float eco_int = i_eu + t_eu;  // economic interdependence
float str_int;  // strategic interdependence
if (i_power > t_power)
{
    bigger = i_power;
    str_int = i_eu * (1.0 - dcon);
}
else
{
    bigger = t_power;
    str_int = t_eu * (1.0 - dcon);
}

int i_major, t_major;  // 0 = not major, 1 = major
i_major = s[pinitiator].get_major();
t_major = s[ptarget].get_major();
int major_size = mp.size();
int minor_size = s.size() - mp.size();
int total_size = s.size();
int real_trade;  // 0 = no real trade, 1 = real trade
if ((vseu.size() > 0) && (s[pinitiator].get_major() == 0) &&
    (s[ptarget].get_major() == 1))
{
    int pii = -1;
    for (int x = 0; x < vseu.size(); x++)
    {
        if (vseu.at(x).id == pinitiator)
        {
            pii = x;
            break;
        }
    }
}

// ==============================================================
// NO ACTION MODULE
// ==============================================================
if (pinitiator_wot == 0)
{
    no_war_freq++;
    for (i = 0; i < c.size(); i++)
    {

    
}
c[i].grow_ura(nr((float)0.01, (float)GR_SD));
c[i].grow_urb(nr((float)0.01, (float)GR_SD));
}
for (i = 0; i < s.size(); i++) s[i].cal_state_power(c);
continue; // Return to the loop
} // End of trade module

// TRADE MODULE

if (pinitiator_wot == 1)
{
  // Trade

  int a_export_ura = -1;
  float a_export_ratio = -1.0;
  float b_export_ratio = -1.0;
  print_screen_2(pinitiator_wot, 3);
  trade(s, c, ERROR_SD, pinitiator, ptarget, real_trade,
        a_export_ura, a_export_ratio, b_export_ratio);
  // a_export_ura: 0 = pinitiator exports urb,
  // 1 = pinitiator exports ura
  // If its value is still -1, an error message will be shown
  if (a_export_ura < 0)
  {
    cerr << "Calculation error for a_export_ura." << endl;
    exit(1);
  }
  trade_freq++;

  // Grow ura and urb, and then recalculate state power level
  //------------------------------------------------------------------------
  for (i = 0; i < c.size(); i++)
  {
    c[i].grow_ura(nr((float)0.01, (float)GR_SD));
    c[i].grow_urb(nr((float)0.01, (float)GR_SD));
  }
  for (i = 0; i < s.size(); i++) s[i].cal_state_power(c);
  continue; // Return to the loop
} // End of trade module

// WAR MODULE

if (pinitiator_wot == 2)
{
  // Alliance declarations (all include ptarget or pinitiator)
  //------------------------------------------------------------------------
  vector<int> defensive_spv;
  vector<int> defensive_alliance;
  vector<int> offensive_spv;
  vector<int> offensive_alliance;
  vector<int> enhanced_spv;
  vector<int> enhanced_alliance;

  // Defensive alliance configuration
  //------------------------------------------------------------------------
defensive_spv.clear();
defensive_spv = util_configure_defensive_alliance
                        (s, c, pinitiator, ptarget, ERROR_SD,
                         WARCOSTMAX, WCDISPAR, REPAR, LVSIGMA);

// Go to war if the target cannot find any available configuration
// Any combination of defensive_spv (including target and others) will
// hold positive expected utilities (according to related functions)
// ---------------------------------------------------------------
if (defensive_spv.size() <= 1)
{
    offensive_part.push_back(pinitiator);
    defensive_part.push_back(ptarget);
}

// Defensive alliance building
// ---------------------------------------------------------------------------
else
{
    vector<int> defensive_alliance;
    defensive_alliance = util_build_defensive_alliance
                        (s, c, pinitiator, ptarget, defensive_spv,
                         ERROR_SD, WARCOSTMAX, WCDISPAR, REPAR, LVSIGMA);

// Go to war if the target fails to build a defensive alliance or
// the expected utilities of alliance (of target and the states who
// accepted the bids) upon recalculation is less than zero, then
// the target gives up forming a defensive alliance
// ---------------------------------------------------------------
if ((defensive_alliance.size() <= 1) ||
    (exp_util_alliance (s, c, ERROR_SD, WARCOSTMAX, WCDISPAR,
                        REPAR, LVSIGMA, ptarget, pinitiator,
                        defensive_alliance, offensive_alliance, true) < 0.0))
{
    offensive_part.push_back(pinitiator);
    defensive_part.push_back(ptarget);
}

// If a defensive alliance is organized, then the initiator configures
// the best combination of offensive_spv in order to maximize the
// expected utilities of war
// ---------------------------------------------------------------------------
else
{
    offensive_spv.clear();
    offensive_spv = util_configure_offensive_alliance
                    (s, c, pinitiator, ptarget, defensive_alliance,
                     ERROR_SD, WARCOSTMAX, WCDISPAR, REPAR, LVSIGMA);

// Stop the loop if there is no candidate for offensive_spv or
// the expected utility of war is less than zero upon the recalculation
// after the bids for offensive alliance; initiator gives up offense
// If additional offensive alliance member is available, then the
// expected utility of war (for initiator) will be positive
// ---------------------------------------------------------------------------
if (offensive_spv.size() <= 1)
\{ 
  \text{offensive\_part}.\text{push\_back}(pinitiator); 
  \text{for} (i = 0; i < \text{defensive\_alliance}.\text{size}(); i++) 
      \text{defensive\_part}.\text{push\_back}(\text{defensive\_alliance}[i]); 
  \text{print\_screen\_2}(2, 0);  \quad // 1 \text{ always represents war} 
  \text{no\_war\_freq}++; 

  //  \text{------------------------------------------------------------------------} 
  // Grow ura and urb, and then recalculate state power level 
  // \text{------------------------------------------------------------------------} 
  \text{for} (i = 0; i < \text{c}.\text{size}(); i++) 
  \{ 
      \text{c}[i].\text{grow\_ura}(n((\text{float})0.01, (\text{float})\text{GR\_SD})); 
      \text{c}[i].\text{grow\_urb}(n((\text{float})0.01, (\text{float})\text{GR\_SD})); 
  \} 
  \text{for} (i = 0; i < \text{s}.\text{size}(); i++) \text{s}[i].\text{cal\_state\_power}(\text{c}); 
  \text{continue}; \quad // \text{Return to the round loop} 
\}

// \text{----------------------------------------------------------------------} 
// \text{offensive\_alliance} 
// \text{----------------------------------------------------------------------} 

\text{offensive\_alliance}.\text{clear}(); 
\text{offensive\_alliance} = \text{util\_build\_offensive\_alliance} 
(\text{s}, \text{c}, \text{pinitiator}, \text{ptarget}, \text{defensive\_alliance}, 
\text{offensive\_spv}, \text{ERROR\_SD}, \text{WARCOSTMAX}, 
\text{WCDISPAR}, \text{REPAR}, \text{LVSIGMA}); 

// \text{----------------------------------------------------------------------} 
// Stop the loop if offensive\_alliance has only the initiator or 
// the expected utility of war is less than after the addition of 
// other alliance members; the initiator gives up offense 
// \text{----------------------------------------------------------------------} 

\text{if} ((\text{offensive\_alliance}.\text{size}() \leq 1)| 
  (\text{exp\_util\_alliance}(\text{s}, \text{c}, \text{ERROR\_SD}, \text{WARCOSTMAX}, \text{WCDISPAR}, 
  \text{REPAR}, \text{LVSIGMA}, \text{pinitiator}, \text{ptarget}, \text{offensive\_alliance}, 
  \text{defensive\_alliance}, \text{true}) < 0.0)) 
\{ 
  \text{for} (i = 0; i < \text{offensive\_alliance}.\text{size}(); i++) 
      \text{offensive\_part}.\text{push\_back}(\text{offensive\_alliance}.\text{at}(i)); 
  \text{for} (i = 0; i < \text{defensive\_alliance}.\text{size}(); i++) 
      \text{defensive\_part}.\text{push\_back}(\text{defensive\_alliance}.\text{at}(i)); 
  \text{print\_screen\_2}(2, 0);  \quad // "2" \text{ represents war} 
  \text{no\_war\_freq}++; 

  // \text{----------------------------------------------------------------------} 
  // Grow ura and urb, and then recalculate state power level 
  // \text{----------------------------------------------------------------------} 
  \text{for} (i = 0; i < \text{c}.\text{size}(); i++) 
  \{ 
      \text{c}[i].\text{grow\_ura}(n((\text{float})0.01, (\text{float})\text{GR\_SD})); 
      \text{c}[i].\text{grow\_urb}(n((\text{float})0.01, (\text{float})\text{GR\_SD})); 
  \} 
  \text{for} (i = 0; i < \text{s}.\text{size}(); i++) \text{s}[i].\text{cal\_state\_power}(\text{c}); 
  \text{continue}; \quad // \text{Return to the round loop} 
\}

// \text{----------------------------------------------------------------------} 
// Target recalculates the EU of war after the formation of offensive
// alliance; if the EU is negative, try more bids; otherwise, go to war
// without further bids for alliance
// -----------------------------------------------------------------------------------------------
if (exp_util_alliance(s, c, ERROR_SD, WARCOSTMAX, WCDISPAR, REPAR, LVSIGMA, ptarget, pinitiator, defensive_alliance, offensive_alliance, true) > 0.0)
{
    for (i = 0; i < offensive_alliance.size(); i++)
        offensive_part.push_back(offensive_alliance.at(i));
    for (i = 0; i < defensive_alliance.size(); i++)
        defensive_part.push_back(defensive_alliance.at(i));
}
// -----------------------------------------------------------------------------------------------
// If the recalculated EU < 0.0, try enhanced_spv
// -----------------------------------------------------------------------------------------------
else
{
    enhanced_spv.clear();
    enhanced_spv = util_configure_enhanced_alliance(s, c, pinitiator, ptarget, defensive_alliance, offensive_alliance, ERROR_SD, WARCOSTMAX, WCDISPAR, REPAR, LVSIGMA);
    // -----------------------------------------------------------------------------------------------
    // Go to war if the target cannot find any further allies, go to war
    // -----------------------------------------------------------------------------------------------
    if (enhanced_spv.size() <= 0)
    {
        for (i = 0; i < offensive_alliance.size(); i++)
            offensive_part.push_back(offensive_alliance.at(i));
        for (i = 0; i < defensive_alliance.size(); i++)
            defensive_part.push_back(defensive_alliance.at(i));
    }
    // -----------------------------------------------------------------------------------------------
    // Enhanced defensive alliance building module
    // -----------------------------------------------------------------------------------------------
    else
    {
        enhanced_alliance.clear();
        enhanced_alliance = util_build_enhanced_alliance(s, c, pinitiator, ptarget, defensive_alliance, offensive_alliance, enhanced_spv, ERROR_SD, WARCOSTMAX, WCDISPAR, REPAR, LVSIGMA);
        // enhanced_alliance may have the same number of members
        // as that of defensive_alliance (or more, but not less)
        for (i = 0; i < offensive_alliance.size(); i++)
            offensive_part.push_back(offensive_alliance[i]);
        for (i = 0; i < enhanced_alliance.size(); i++)
            defensive_part.push_back(enhanced_alliance[i]);
        // (End) enhanced_alliance
    }
    // (End) enhanced_spv
    } // (End) offensive_spv and offensive_alliance
    // (End) defensive_alliance
    war_freq++;
// -----------------------------------------------------------------------------------------------
// War and reparations module (Functions defined in War.cpp)
// -----------------------------------------------------------------------------------------------

251

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
float offensive_power = 0.0;
float defensive_power = 0.0;
float t; // power ratio between offensive and defensive parts
for (int to = 0; to < offensive_part.size(); to++)
    offensive_power += s[offensive_part[to]].get_state_power();
for (int td = 0; td < defensive_part.size(); td++)
    defensive_power += s[defensive_part[td]].get_state_power();
t = offensive_power / defensive_power;
float lv = likelihood_of_victory(t, LVSIGMA);

float warcells = 0.0;
for (int i = 0; i < offensive_part.size(); i++)
    warcells += s[offensive_part[i]].get_sc_size();
for (int i = 0; i < defensive_part.size(); i++)
    warcells += s[defensive_part[i]].get_sc_size();
float pinitiator_power = s[pinitiator].get_state_power();
float ptarget_power = s[ptarget].get_state_power();

float minn = warcost_minimum(WARCOSTMAX, WCDISPAR);
float warcostfactor = cal_warcost(offensive_power, defensive_power, WARCOSTMAX);

float rnd = uniform_random(); // To be compared with lv and lt

vector<int> victor_part; // Vector variables for war results
vector<int> defeated_part;
vector<state_power> vsp; // Sorted list of victors + powers
vector<int> victor_list; // Sorted list of victors
vector<float> victor_power; // Sorted list of victor power
float proportion; // Proportion: territories to transfer
float cell_transfer_ratio;
float templ; // templ for integer
double temp2; // temp2 for fraction by modf()
float vp_total = 0.0; // Victor coalition's total power
float indem_total = 0.0;
vector<float> indem_vector;
int cell_loss; // Number of cells to be transferred
vector<int> cell_list; // All cells from the defeated
vector<float> victor_power_ratio; // Power ratio among victors
vector<int> precell_claim; // Integer version: victor_power_ratio
vector<int> intcell_claim; // Integer vector of cell claim
vector<int> cell_claimants; // Vector of states claiming cells
float indem_lv;
vector<int> pool_new_states; // Pool of newly created states
int claimant; // Current cell claimant id
int how_many; // Current cell claimant's request
int original_loser_size;
vector<int> contiguous_cells;
vector<int> fragmented;
vector< vector<int> > v_conti_blocks;
vector<int> main_block;

// CASE 1: INITIATOR WINS
// INITIATOR WINS, TARGET LOSES
if (lv > rnd)
{
    initiator_wins++;
    //print_who_wins(fout0, 1);
    print_screen_2(pinitiator_wot, 1);

    // WHO WINS?
    vector_part = offensive_part;
    defeated_part = defensive_part;
    for (i = 0; i < vector_part.size(); i++)
    {
        state_power sp;
        sp.state = vector_part[i];
        s[vector_part[i]].cal_state_power(c);
        sp.power = s[vector_part[i]].get_state_power();
        vsp.push_back(sp);
    }
    sort(vsp.begin(), vsp.end(), by_state_power);
    for (i = 0; i < vsp.size(); i++)
    {
        victor_list.push_back(vsp[i].state);
        victor_power.push_back(vsp[i].power);
        vp_total += vsp[i].power;
    }

    // WAR COST PAYMENT
    for (i = 0; i < vector_part.size(); i++)
        s[vector_part[i]].reduce_state_power(warcostfactor, c);
    for (i = 0; i < defeated_part.size(); i++)
        s[defeated_part[i]].reduce_state_power(warcostfactor, c);

    // INDEMNITIES CALCULATION
    for (i = 0; i < defeated_part.size(); i++)
    {
        float indem_id;
        indem_id = s[defeated_part[i]].get_state_power() * REPAR;
        indem_vector.push_back(indem_id);
        indem_total += indem_id;
    }

    // CELLS CALCULATION
}
proportion = lv;
cell_transfer_ratio = proportion * s[ptarget].get_sc_size();
if (cell_transfer_ratio < 1.0) cell_loss = 1;
else {
    templ = (float) modf(cell_transfer_ratio, &temp2);
cell_loss = (int) temp2; // Casting
    if (templ >= 0.5) // Rounding up
        cell_loss++;
}
if (s[ptarget].get_sc_size() <= cell_loss)
cell_loss = s[ptarget].get_sc_size();
for (i = 0; i < victor_power.size(); i++)
    victor_power_ratio.push_back(victor_power.at(i) / vp_total);
precell_claim = cal_precell_claim(cell_loss, victor_power_ratio);
intcell_claim = cal_cell_claim(precell_claim);
intcell_claimants = cal_cell_claimants(intcell_claim, victor_list, s);

// INDEMNITIES TRANSFERS
// -----------------------------------------------------------------------------
for (i = 0; i < defeated_part.size(); i++)
{
    s[defeated_part[i]].subtract_state_power(indem_vector[i], c);
s[defeated_part[i]].cal_state_power(c);
}
for (i = 0; i < victor_part.size(); i++)
{
    indem_iv = (s[victor_part[i]].get_state_power() / vp_total) * indem_total;
s[victor_part[i]].add_state_power(indem_iv, c);
s[victor_part[i]].cal_state_power(c);
}

// CASE 1 CELLS TRANSFERS
// -----------------------------------------------------------------------------
while (cell_claimants.size() > 0)
{
    claimant = s[cell_claimants.back()].get_id();
    how_many = intcell_claim.back();
    bool cut = false; // whether loser is cut or not
    original_loser_size = s[ptarget].get_sc_size();
    contiguous_cells.clear();
    for (i = 0; i < s[ptarget].get_sc_size(); i++)
        if (s[claimant].find_snc(c, s[ptarget].get_sc(i)))
            contiguous_cells.push_back(s[ptarget].get_sc(i));
    // If the loser transfers all cells to one claimant
    if ((contiguous_cells.size() > 0) &&
        (cell_claimants.size() == 1) &&
        (how_many == original_loser_size))
    {
        for (i = 0; i < original_loser_size; i++)
        {

254
int pcell = s[ptarget].get_sc(0);
    cell_transfer3
        (s, c, ptarget, claimant, pcell);
    break;
}
while (how_many > 0)
{
    contiguous_cells.clear();
    for (i = 0; i < s[ptarget].get_sc_size(); i++)
        if (s[claimant].find_snc(c, s[ptarget].get_sc(i)))
            contiguous_cells.push_back(s[ptarget].get_sc(i));
    if (contiguous_cells.size() == 1)
    {
        cell_transfer1(s, c, ptarget, claimant,
                        contiguous_cells.back(), how_many);
        continue;
    }
    else if ((contiguous_cells.size() > 1) || (!cut))
    {
        make_nonsplit(s, c, claimant, ptarget,
                      contiguous_cells, cut);
        if (contiguous_cells.size() == 1)
        {
            cell_transfer1(s, c, ptarget, claimant,
                            contiguous_cells.back(), how_many);
            continue;
        }
    }
    else if (contiguous_cells.size() > 0)
    {
        make_winner_compact
            (pinitiator, c, contiguous_cells);
        if (contiguous_cells.size() == 1)
        {
            cell_transfer1(s, c, ptarget, claimant,
                            contiguous_cells.back(), how_many);
            continue;
        }
    }
    else if (contiguous_cells.size() > 0)
    {
        make_loser_compact
            (ptarget, c, contiguous_cells);
        if (contiguous_cells.size() == 1)
        {
            cell_transfer1(s, c, ptarget, claimant,
                            contiguous_cells.back(), how_many);
            continue;
        }
    }
    else // if contiguous_cells.size() == 0
    {
        cerr << "Exception: ERROR #1 CASE 1"
             << endl;
        exit(1);
    }
}
else if (contiguous_cells.size() == 0)
{
    cerr << "Exception: ERROR #2 CASE 1" << endl;
    exit(1);
}
else
{
    cut = true;
    continue;
}
else if ((contiguous_cells.size() > 1) && (!cut))
    transfer_first_cell(contiguous_cells, s, c, ptarget, claimant, how_many);
else if (!contiguous_cells.size())
    break;
}  // End while (how_many > 0)

// Remove the "claimant" from the list if how_many or cell_loss = 0
// -----------------------------------------------------------------------------------------------

// Rearrange the target state cells in case it is split
// -----------------------------------------------------------------------------------------------

if (s[ptarget].get_sc_size() > 0)
{

    // If the state is contiguous without a capital, make one
    if (s[ptarget].cells_contiguous(s, c))
    {
        if (!s[ptarget].has_capital())
            c[s[ptarget].get_sc(0)].set_whether_capital(1);
    }
    else // If the cells are not contiguous
    {
        for (i = 0; i < s[ptarget].get_sc_size(); i++)
            fragmented.push_back(s[ptarget].get_sc(i));
        for (i = 0; i < fragmented.size(); i++)
            c[fragmented[i]].set_whether_capital(0);
        v_conti_blocks = make_contiguous_blocks(c, fragmented);
        main_block = choose_main_block(v_conti_blocks);
        c[main_block[0]].set_whether_capital(1);  // set a capital

    // Now convert the remaining contiguous cell blocks to new states
    // -----------------------------------------------------------------------------------------------

    for (i = 0; i < v_conti_blocks.size(); i++)
    {
        if (v_conti_blocks[i] == main_block) continue;
        new_state_id++;
        new_state_number++;
        new_state_transfer2(new_state_id, v_conti_blocks[i].at(0), s, c, ptarget, META_SD);
        int ps = s.back().get_id();
        if (v_conti_blocks[i].size() > 1)
for (int j = 1; j < v_conti_blocks[i].size(); j++)
    {
        int more = v_conti_blocks[i].at(j);
        cell_transfer2(s, c, ptarget, ps, more);
    }
// End for
}  // End else
}  // End if
}  // End CASE

// CASE 2: TARGET WINS
// TARGET WINS, INITIATOR LOSES
else
{
target_wins++;
print_screen_2(pinitiator_wot, 2);

// WHO WINS?
victim_part = defensive_part;
defeated_part = offensive_part;
for (i = 0; i < victor_part.size(); i++)
    {
        state_power sp;
        sp.state = victor_part[i];
        s[victor_part[i]].cal_state_power(c);
        sp.power = s[victor_part[i]].get_state_power();
        vsp.push_back(sp);
    }
    sort(vsp.begin(), vsp.endO, by_state_power);
    for (i = 0; i < vsp.size(); i++)
    {
        victor_list.push_back(vsp[i].state);
        victor_power.push_back(vsp[i].power);
        vp_total += vsp[i].power;
    }

// WAR COST PAYMENT
for (i = 0; i < victor_part.size(); i++)
    s[victor_part[i]].reduce_state_power(warcostfactor, c);
for (i = 0; i < defeated_part.size(); i++)
    s[defeated_part[i]].reduce_state_power(warcostfactor, c);

// INDEMNITIES CALCULATION
for(i = 0; i < defeated_part.size(); i++)
    {
        float indem_id;
        indem_id = s[defeated_part[i]].get_state_power() * REPAR;
        indem_vector.push_back(indem_id);
        indem_total += indem_id;
    }
proportion = 1.0 - lv;
cell_transfer_ratio = proportion
    * s[pinitiator].get_sc_size();
if (cell_transfer_ratio < 1.0) cell_loss = 1;
else
{
    templ = (float) modf(cell_transfer_ratio, &temp2);
cell_loss = (int) temp2; // Casting
    if (templ >= 0.5) // Rounding up
        cell_loss++;
}
if (s[pinitiator].get_sc_size() <= cell_loss)
cell_loss = s[pinitiator].get_sc_size();
for (i = 0; i < victor_power.size(); i++)
victor_power_ratio.push_back(victor_power.at(i)/vp_total);
precell_claim = cal_precell_claim
    (cell_loss, victor_power_ratio);
intcell_claim = cal_cell_claim(precell_claim);
intcell_claimants = cal_cell_claimants
    (intcell_claim, victor_list, s);

for (i = 0; i < defeated_part.size(); i++)
{
    s[defeated_part[i]].subtract_state_power
        (indem_vector[i], c);
s[defeated_part[i]].cal_state_power(c);
}
for (i = 0; i < victor_part.size(); i++)
{
    indem_iv = (s[victor_part[i]].get_state_power
        / vp_total) * indem_total;
s[victor_part[i]].add_state_power(indem_iv, c);
s[victor_part[i]].cal_state_power(c);
}

while (cell_claimants.size() > 0)
{
    claimant = s[cell_claimants.back()].get_id();
    how_many = intcell_claim.back(); // whether loser is cut or not
    bool cut = false; // original_loser_size = s[pinitiator].get_sc_size();
    contiguous_cells.clear();
    for (i = 0; i < s[pinitiator].get_sc_size(); i++)
    if (s[claimant].find_snc
        (c, s[pinitiator].get_sc(i))
        contiguous_cells.push_back
            (s[pinitiator].get_sc(i));

    // If the loser transfers all cells to one claimant
if ((contiguous_cells.size() > 0) &&
    (cell_claimants.size() == 1) &&
    (how_many == original_loser_size))
for (i = 0; i < original_loser_size; i++)
{
    int pcell = s[pinitiator].get_sc(0);
    cell_transfer3(s, c, pinitiator, claimant, pcell);
    break;
}

while (how_many > 0)
{
    contiguous_cells.clear();
    for (i = 0; i < s[pinitiator].get_sc_size(); i++)
    if (s[claimant].find_snc(c, s[pinitiator].get_sc(i)))
        contiguous_cells.push_back(s[pinitiator].get_sc(i));
    if (contiguous_cells.size() == 1)
    {
        cell_transfer1(s, c, pinitiator, claimant, contiguous_cells.back(), how_many);
        continue;
    }
    else if ((contiguous_cells.size() > 1) && (!cut))
    {
        make_nonsplit(s, c, claimant, pinitiator, contiguous_cells, cut);
        if (contiguous_cells.size() == 1)
        {
            cell_transfer1(s, c, pinitiator, claimant, contiguous_cells.back(), how_many);
            continue;
        }
    }
    else if (contiguous_cells.size() > 0)
    {
        make_winner_compact(ptarget, c, contiguous_cells);
        if (contiguous_cells.size() == 1)
        {
            cell_transfer1(s, c, pinitiator, claimant, contiguous_cells.back(), how_many);
            continue;
        }
    }
    else if (contiguous_cells.size() > 0)
    {
        make_loser_compact(pinitiator, c, contiguous_cells);
        if (contiguous_cells.size() == 1)
        {
            cell_transfer1(s, c, pinitiator, claimant, contiguous_cells.back(), how_many);
            continue;
        }
    }
    else if (contiguous_cells.size() > 1)
    {
        transfer_first_cell(contiguous_cells, s, c, pinitiator, claimant, how_many);
        continue;
    }
}
else // contiguous_cells.size() == 0
{
    cerr << "Exception: ERROR #1 "
    << "CASE 2" << endl;
    exit(1);
}
else // contiguous_cells.size() == 0
{
    cerr << "Exception: ERROR #2 "
    << "CASE 2" << endl;
    exit(1);
}
else
{
    cut = true;
    continue;
}
else if ((contiguous_cells.size() > 1) && (cut))
    transfer_first_cell(contiguous_cells, s, c, pinitiator, claimant, how_many);
else break; // if no contiguous_cells

// Remove the "claimant" from the list if how_many or cell_loss = 0
// Remove the "claimant" from the list if how_many or cell_loss = 0
// Remove the "claimant" from the list if how_many or cell_loss = 0
cell_claimants.pop_back();
intcell_claim.pop_back();
} // End while (cell_claimants.size() > 0) loop

// Rearrange the initiator state cells in case it is split
// Rearrange the initiator state cells in case it is split
// Rearrange the initiator state cells in case it is split
if (s[pinitiator].get_sc_size() > 0)
{
    // If the state is contiguous without a capital, make one
    if (s[pinitiator].cells_contiguous(s, c))
    {
        if (!s[pinitiator].has_capital())
            c[s[pinitiator].get_sc(0)].set_whether_capital(1);
    }
    else // If the cells are not contiguous
    {
        for (i = 0; i < s[pinitiator].get_sc_size(); i++)
            fragmented.push_back(s[pinitiator].get_sc(i));
        for (i = 0; i < fragmented.size(); i++)
            c[fragmented[i]].set_whether_capital(0);
        v_conti_blocks = make_contiguous_blocks(c, fragmented);
        main_block = choose_main_block(v_conti_blocks);
        c[main_block[0]].set_whether_capital(1); // set a capital
    }
}
else // contiguous_cells.size() == 0
{
    cerr << "Exception: ERROR #1 "
    << "CASE 2" << endl;
    exit(1);
}
else // contiguous_cells.size() == 0
{
    cerr << "Exception: ERROR #2 "
    << "CASE 2" << endl;
    exit(1);
}
}
new_state_number++; 
new_state_transfer2(new_state_id, 
v_conti_blocks[i].at(0), s, c, pinitiator, META_SD); 
int ps = s.back().get_id(); 
if (v_conti_blocks[i].size() > 1) 
{
    for (int j = 1; j < v_conti_blocks[i].size(); j++)
    {
        int more = v_conti_blocks[i].at(j);
        cell_transfer2(s, c, pinitiator, ps, more);
    }
}
// End for 
// End else 
// End if 
// End CASE 

// Power adjustment phase 
for (i = 0; i < c.size(); i++)
{
    c[i].grow_ura(nr((float)0.01, (float)GR_SD));
    c[i].grow_urb(nr((float)0.01, (float)GR_SD));
}
for (i = 0; i < s.size(); i++) s[i].cal_state_power(c);
// End of war module 
// End of round loop 

// Print out the dependent variable data to fout1 
print_result(fout1, new_state_number, trade_freq, no_war_freq, 
war_freq, target_wins, Initiator_wins, 
round - 1, final_state_number); 
// End of run loop 
fout1.close();
fout2.close();
fout3.close(); 
return 0; 
// End of main()
```cpp
#ifndef CELL_H
#define CELL_H

#include <vector>
#include <list>
#include "Param.h"
#include "State.h"

using namespace std;

class State; // Forward declaration

class Cell
{
    private:
        int id;       // 0 - 97
        int position; // 1 - 98
        float ura;
        float urb;
        int which_state; // To which this cell belongs?
        int whether_capital;
        vector<int> neighbor_cell; // The vector of neighbor cells

    public:
        // Constructors and destructors
        Cell();
        Cell(int p); // p+1 as position
        Cell(const Cells rc); // Copy constructor
        Cells operator=(const Cells rc); // Assignment operator
        ~Cell();

        // Retrieval of position value
        int get_id(); // 0 - 97
        int get_position(); // 1 - 98

        // Neighbor setting
        void set_neighbor_cell();
        int get_neighbor_cell_size();
        int get_neighbor_cell(int a); // 0 - 97
        int get_neighbor_cell_position(int i, vector<Cell> & vc); // 1 - 98
        bool is_neighbor_cell(vector<Cell> & vc, int a); // Is vc[a] a neighbor cell of this cell?
        // Set the vector of contiguous cell vectors at each level of distance

};

#endif
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
vector< vector<int> > set_contiguous_vv
  (vector<State> & vs, vector<Cell> & vc);
// Whether a cell "pc" is contiguous to this cell or not
bool is_contiguous(vector<vector<int> > contiguous_vv, int a);

// Resources and metabolism setting
// ==================================================
float get_ura() {  return ura; }
float get_urb() {  return urb; }
void set_ura(float a);
void set_urb(float a);
void grow_ura(float gr);
void grow_urb(float gr);
void add_ura(float n);
void add_urb(float n);
void reduce_ura(float n);
void reduce_urb(float n);

// Cell's state identity and capital
// ==============================================================
int get_which_state() { return which_state; }
int get_whether_capital() { return whether_capital; }
void set_which_state(int a);  // Assigns the cell to a state vs[a]
void set_whether_capital(int wc);
string get_which_state_name(vector<State> & vs);

#endif  // _CELL_H_
// Definition of the Cell class functions

#include <iostream>
#include <iomanip>
#include "Cell.h"

using namespace std;

// Constructors and destructors

Cell::Cell() : id(-1), position(-1), ura(0.0), urb(0.0), which_state(-1), whether_capital(1)
{
    neighbor_cell.clear();
    for (int i = 0; i < 6; i++)
        neighbor_cell.push_back(-1);
}

Cell::Cell(int p) : id(p), position(p+1), ura(0.0), urb(0.0), which_state(-1), whether_capital(1)
{
    neighbor_cell.clear();
    for (int i = 0; i < 6; i++)
        neighbor_cell.push_back(-1);
}

Cell::Cell(const Cells rc) : id(rc.id), position(rc.position), ura(0.0), urb(0.0), which_state(rc.which_state), whether_capital(rc.whether_capital), neighbor_cell(rc.neighbor_cell)
{
    neighbor_cell.clear();
    for (int i = 0; i < 6; i++)
        neighbor_cell.push_back(-1);
}

Cells Cell::operator=(const Cells rc)
{
    id = rc.id;
    position = rc.position;
    ura = rc.ura;
    urb = rc.urb;
    which_state = rc.which_state;
    whether_capital = rc.whether_capital;
    neighbor_cell = rc.neighbor_cell;
    return *this;
}
Cell::~Cell()
{
    neighbor_cell.clear();
}

// Neighbor setting
// Cases of odd numbers
void Cell::set_neighbor_cell()
{
    // Cases of odd numbers
    if (position%2 == 1) // Odd numbers
    {
        if (position/14 == 0) // 1,3,5,7,9,11,13 (1st row)
        {
            neighbor_cell[0] = -1;
            neighbor_cell[1] = -1;
            neighbor_cell[2] = -1;
            neighbor_cell[3] = position + 1 - 1;
            neighbor_cell[4] = position + 14 - 1;
        }
        else // 3,5,7,9,11,13
        {
            neighbor_cell[0] = position - 1 - 1;
        }
    }
    else // The remaining rows
    {
        if (position/14 == 6) // 85,87,89,91,93,95,97
        {
            neighbor_cell[1] = position - 14 - 1;
            neighbor_cell[2] = position - 13 - 1;
            neighbor_cell[3] = position + 1 - 1;
            neighbor_cell[4] = -1;
        }
        else // All other odd id's
        {
            neighbor_cell[0] = position - 15 - 1;
            neighbor_cell[5] = position - 1 - 1;
        }
    }

else // Even numbers
{
    if (position%14 == 0) // 14,28,42,56,70,84,98
    {
        neighbor_cell[0] = position - 1 - 1;
        neighbor_cell[2] = -1;
        neighbor_cell[3] = -1;
        if (position/14 == 1) // 14
        {
            neighbor_cell[1] = -1;
            neighbor_cell[4] = position + 14 - 1;
            neighbor_cell[5] = position + 13 - 1;
        }
    }
    else
    {
        if (position/14 == 7) // 98
        {
            neighbor_cell[1] = position - 14 - 1;
            neighbor_cell[4] = -1;
            neighbor_cell[5] = -1;
        }
        else // 28,42,56,70,84
        {
            neighbor_cell[1] = position - 14 - 1;
            neighbor_cell[4] = position + 14 - 1;
            neighbor_cell[5] = position + 13 - 1;
        }
    }
}
else // Columns except the rightmost
{
    neighbor_cell[0] = position - 1 - 1;
    neighbor_cell[2] = position + 1 - 1;
    if (position/14 == 0) // 1st row (2,4,6,8,10,12)
    {
        neighbor_cell[1] = -1;
        neighbor_cell[3] = position + 15 - 1;
        neighbor_cell[4] = position + 14 - 1;
        neighbor_cell[5] = position + 13 - 1;
    }
    else
    {
        if (position/14 == 6) // 86,88,90,92,94,96
        {
            neighbor_cell[1] = position - 14 - 1;
            neighbor_cell[3] = -1;
            neighbor_cell[4] = -1;
            neighbor_cell[5] = -1;
        }
        else
        {
            neighbor_cell[1] = position - 14 - 1;
            neighbor_cell[3] = position + 15 - 1;
            neighbor_cell[4] = position + 14 - 1;
        }
    }
}

266
int Cell::get_neighbor_cell_size()
{
    int neighbor_number = 0;
    for (int i = 0; i < 6; i++)
    {
        if (neighbor_cell[i] != -1)
            neighbor_number++;
    }
    return neighbor_number;
}

int Cell::get_neighbor_cell(int a)
{
    return neighbor_cell[a];
}

int Cell::get_neighbor_cell_position(int a, vector<Cell> & vc)
{
    int index = this->neighbor_cell[a];
    return vc[index].get_position();
}

// Is vc[a] a neighbor cell of this cell?
bool Cell::is_neighbor_cell(vector<Cell> & vc, int a)
{
    for(int i = 0; i < this->get_neighbor_cell_size(); i++)
        if (this->get_neighbor_cell_position(i, vc) == vc[a].position)
            return true;
    return false;
}

// Set the vector of contiguous cell vectors at each level of distance
vector<vector<int> > Cell::set_contiguous_w(vector<State> & vs, vector<Cell> & vc)
{
    // Container parameters
    vector<vector<int> > contiguous_w; // return vector
    vector<int> conti; // already found int vector
    vector<int> previous_v; // previous vector for rounds
    int ps = this->get_which_state();
    int total_cells = vs[ps].get_sc_size();

    // Make distance 0 contiguous cell vector (itself)
    // already found vector
    vector<int> v0;
    v0.push_back(this->get_id());
    conti.push_back(this->get_id());
    contiguous_w.push_back(v0);
    previous_v.push_back(v0);

    // End of even numbers
    // End of function definition
Make a contiguous cell vector and push back to contiguous_vv

for (int i = 0; i < previous_v.size(); i++)
{
    vector<int> temp; temp.clear();
    for (int j = 0; j < 6; j++)
    {
        int n = vc[previous_v[i]].get_neighbor_cell(j);
        if (n < 0 ) continue;
        iter1 = find(conti.begin(), conti.end(), n);
        iter2 = find(previous_v.begin(), previous_v.end(), n);
        if ((vc[n].get_which_state() != ps)I (iter1 != conti.end())||
            (iter2 != previous_v.end()))
            continue;
        temp.push_back(n);
        conti.push_back(n);
        contiguous_w.push_back(temp);
        previous_v = temp;
    }
    if (temp.size() == 0) return contiguous_vv; // no cell contiguous
    if (total_cells == conti.size()) return contiguous_vv;
}
return contiguous_vv;

// Whether a cell "pc" is contiguous to this cell or not
bool Cell::is_contiguous(vector<vector<int>> contiguous_vv, int pc)
{
    for (int i = 0; i < contiguous_vv.size(); i++)
    {
        int n = contiguous_vv.at(i).size();
        for (int j = 0; j < n; j++)
            if (pc == contiguous_vv.at(i).at(j))
                return true;
    }
    return false;
}

// Resources and metabolism setting
void Cell::set_ura(float a) { ura = a; }
void Cell::set_urb(float a) { urb = a; }
void Cell::grow_ura(float gr) { ura = ura * (1.0 + gr); }
void Cell::grow_urb(float gr) { urb = urb * (1.0 + gr); }
void Cell::add_ura(float n) { ura = ura + n; }
void Cell::add_urb(float n) { urb = urb + n; }
void Cell::reduce_ura(float n)
{
    if (this->get_ura() < n)
        ura = MINIMUM_UR;
    else
        ura = ura - n;
}
void Cell::reduce_urb(float n)
{
    if (this->get_urb() < n)
urb = MINIMUM_UR;
else
    urb = urb - n;
}

// Cell's state identity and capital
// ---------------------------------------------------------------------------
void Cell::set_which_state(int a) // Assigns the cell to a state vs[a]
{
    which_state = a;
}

void Cell::set_whether_capital(int wc)
{
    whether_capital = wc;
}

string Cell::get_which_state_name(vector<State> & vs)
{
    string str = vs[this->which_state].get_name();
    return str;
}
// **************************************************************************************
// Expected.h
// Function definitions regarding expected utility calculation
// **************************************************************************************

#ifndef _EXPECTED_H_
#define _EXPECTED_H_

#include <vector>
#include "State.h"
#include "Cell.h"
#include "Prints.h"

using namespace std;

// Calculate the total power of states (with error, except the observer)
float states_power(vector<State> & vs, vector<int> vi, float ERROR_SD);

// Decides whether a state is the strongest in a vector of states
bool is_assumed_strongest(int a, vector<int> vi, vector<State> & vs, float ERROR_SD);

// Calculate the minimum
float warcost_minimum(float WARCOSTMAX, float WCDISPAR);

// War cost factor calculation
float cal_warcost(float a_power, float b_power, float WARCOSTMAX);

// Expected utility of state A if it attacks state B
float exp_util_war_state(vector<State> & vs, vector<Cell> & vc, float ERROR_SD,
float WARCOSTMAX, float WCDISPAR, float REPAR,
float LVSIGMA, int a, int b, bool is_a_leader,
bool is_b_leader, float lv, float a_power, float b_power);

// Expected utility of state A toward a group of states
// va includes state a, vb includes state b
// a is this state, b is the leading enemy state (initiator or target)
float exp_util_alliance(vector<State> & vs, vector<Cell> & vc, float ERROR_SD,
float WARCOSTMAX, float WCDISPAR, float REPAR,
float LVSIGMA, int a, int b, vector<int> va,
vector<int> vb, bool is_a_leader);

// Setting the trade unit size according to two powers
float set_trade_unit(float a, float b);
float estimate_state_power(float ura, float urb, float ma, float mb);

float eu_trade_state(vector<State> & vs, vector<Cell> & vc, int a, int b, float ERROR_SD);

void trade(vector<State> & vs, vector<Cell> & vc, float ERROR_SD, int a, int b, int & real_trade, int & a_export_ura, float & a_export_ratio, float & b_export_ratio);

vector<int> util_configure_defensive_alliance(vector<State> & vs, vector<Cell> & vc, int pinitiator, int ptarget, float ERROR_SD, float WARCOSTMAX, float WCDISPAR, float REPAR, float LVSIGMA);

vector<int> util_build_defensive_alliance(vector<State> & vs, vector<Cell> & vc, int pinitiator, int ptarget, vector<int> defensive_spv, float ERROR_SD, float WARCOSTMAX, float WCDISPAR, float REPAR, float LVSIGMA);

vector<int> util_configure_offensive_alliance(vector<State> & vs, vector<Cell> & vc, int pinitiator, int ptarget, vector<int> defensive_alliance, float ERROR_SD, float WARCOSTMAX, float WCDISPAR, float REPAR, float LVSIGMA);

vector<int> util_build_offensive_alliance(vector<State> & vs, vector<Cell> & vc, int pinitiator, int ptarget, vector<int> defensive_alliance, vector<int> offensive_spv, float ERROR_SD, float WARCOSTMAX, float WCDISPAR, float REPAR, float LVSIGMA);

vector<int> util_configure_enhanced_alliance(vector<State> & vs, vector<Cell> & vc, int pinitiator, int ptarget, vector<int> defensive_alliance, vector<int> offensive_alliance, float ERROR_SD, float WARCOSTMAX, float WCDISPAR, float REPAR, float LVSIGMA);
vector<int> util_build_enhanced_alliance(vector<State> & vs,
    vector<Cell> & vc, int pinitiator, int ptarget,
    vector<int> defensive_alliance, vector<int> offensive_alliance,
    vector<int> enhanced_spv, float ERROR_SD, float WARCOSTMAX,
    float WCDISPAR, float REPAR, float LV$\Sigma$GA);
// Expected.cpp
// Function definitions regarding expected utility calculation

#include <iostream>
#include <cmath>
#include "Param.h"
#include "Mathfunc.h"    // likelihood_of_victory()
#include "Prewar.h"      // states_power(), is_assumed_strongest()
#include "Expected.h"

using namespace std;

// Calculate the total power of states (with error, except the observer)
float states_power(vector<State> & vs, vector<int> vi, float ERROR_SD)
{
    if (vi.size() == 0)
    {
        cerr << "This vector of state numbers is empty in states_power()." << endl;
        exit(1);
    }
    float total_power = 0.0;
    for (int i = 0; i < vi.size(); i++)
        total_power += vs[vi[i]].get_state_power() * (1.0 + nr(0.0, ERROR_SD));
    return total_power;
}

// Decides whether a state is the strongest in a vector of states
bool is_assumed_strongest(int a, vector<int> vi, vector<State> & vs, float ERROR_SD)
{
    // vi does not include state a
    float a_power = vs[a].get_state_power() * nr(0.0, ERROR_SD);
    float strongest_power = a_power;
    for(int i = 0; i < vi.size(); i++)
    {
        if (vi[i] == a) continue;
        float i_power = vs[vi[i]].get_state_power() * nr(0.0, ERROR_SD);
        if (i_power > strongest_power)
            strongest_power = i_power;
    }
    if (strongest_power == a_power) return true;
    else return false;
}

// Calculate the minimum
float warcost_minimum(float WARCOSTMAX, float WCDISPAR)
{
    float random_number = random_float((float)0.0, (float)1.0);
    }
if (random_number * WARCOSTMAX > WCDISPAR)
    return WCDISPAR;
else
    return random_number * WARCOSTMAX;

// War cost factor calculation
float cal_warcost(float a_power, float b_power, float WARCOSTMAX)
{
    // lsr is the ratio of the power of the larger side to the total
    float lsr, warcostfactor, larger_side_power;
    float both_sides_power = a_power + b_power;
    if (a_power > b_power) larger_side_power = a_power;
    else larger_side_power = b_power;
    lsr = larger_side_power / both_sides_power;
    warcostfactor = (1.0 - (lsr-0.5)/0.5) * WARCOSTMAX;
    return warcostfactor;
}

// Expected utility of state A if it attacks state B
float exp_util_war_state(vector<State> & vs, vector<Cell> & vc, float ERROR_SD,
        float WARCOSTMAX, float WCDISPAR, float REPAR, float LVSIGMA,
        int a, int b, bool is_a_leader, bool is_b_leader, float lv,
        float a_power, float b_power)
{
    // Variable list
    // --------------------------------------------------------------------------------------------
    float expected_utility; // return value
    float gains; // assumed gains
    float losses; // assumed losses
    float a_cells;
    int a_cells_lose;
    int a_cells_win;
    float a_indem_lose;
    float a_indem_win;
    float a_warcost;

    float b_cells;
    float power_ratio;
    float warcostfactor;
    float minn; // minn = min(rnd(0.0, 1.0) 'WARCOSTMAX, WCDISPAR)
    if (b_power < 0.0) b_power = MINIMUM_POWER;
    power_ratio = a_power / b_power;
    minn = warcost_minimum(WARCOSTMAX, WCDISPAR);
    warcostfactor = cal_warcost(a_power, b_power, WARCOSTMAX);
    lv = likelihood_of_victory(power_ratio, LVSIGMA);
    b_cells = vs(b].get_sc_size();
    a_indem_win = b_power * (1.0 - warcostfactor) * REPAR;
    // --------------------------------------------------------------------------------------------
    // Gains
    // --------------------------------------------------------------------------------------------

if (is_b_leader) // if b is a leader, b will lose cells
{
  if (b_cells == 1) gains = b_power * (1.0 - warcostfactor);
  else
  {
    a_cells_win = round(b_cells * lv);
    gains = a_indem_win + ((float) a_cells_win
      / (float) b_cells * (b_power * (1.0 - warcostfactor)
        - a_indem_win));
  }
} else // if b is not a leader
  gains = a_indem_win;

// Cost
if (a_power > b_power) a_warcost = a_power * (warcostfactor - minn);
else a_warcost = a_power * (warcostfactor + minn);

// Losses
a_cells = vs[a].get_sc_size();
a_indem_lose = (a_power - a_warcost) * REPAR;
if (is_a_leader) // if a is a leader, a will lose cell
{
  if (a_cells == 1) losses = a_power - a_warcost;
  else
  {
    a_cells_lose = round(a_cells * (1.0 - lv));
    losses = a_indem_lose + ((float) a_cells_lose
      / (float) a_cells * (a_power - a_indem_lose - a_warcost));
  }
} else // if a is not a leader
  losses = a_indem_lose;

// Expected utility
expected_utility = lv * gains - ((1.0 - lv) * losses) - a_warcost;
return expected_utility;

float exp_util_alliance(vector<State> & vs, vector<Cell> & vc, float ERROR_SD,
  float WARCOSTMAX, float WCDISPAR, float REPAR, float LVSIGMA,
  int a, int b, vector<int> va, vector<int> vb, bool is_a_leader)
{
  // Variable list
  float expected_utility; // return value
  float gains;
  float losses;
  
  // Expected utility of state A toward a group of states
  // va includes state a, vb includes state b
  // a is this state, b is the leading enemy state (initiator or target)
  // --------------------------------------------------------------
float lv;
float a_power;
int a_cells;
int a_cells_win;
int a_cells_lose;
float a_indem_win;
float a_indem_lose;
float a_warcost;
float va_power;
float a_weight; // weight of a in total power of alliance
int va_cells_win; // assumed number of cells the alliance will get
float va_indem_win;
float warcostfactor;
float minn;
float b_power;
float b_warcost;
int b_cells;
float vb_power;
float va_warcost, vb_warcost;
float power_ratio;

// Vector initialization
// ---------------------------------------------------------------------------
if (va.size() == 0)   
    va.push_back(a);  // Make va holding at least a
if (vb.size() == 0)   
    vb.push_back(b);  // Make vb holding at least b

// Basic calculation
// ---------------------------------------------------------------------------
a_power = vs[a].get_state_power() * (1.0 + nr(0.0, ERROR_SD));
b_power = vs[b].get_state_power() * (1.0 + nr(0.0, ERROR_SD));
a_cells = vs[a].get_sc_size();
b_cells = vs[b].get_sc_size();
va_power = states_power(vs, va, ERROR_SD);
vb_power = states_power(vs, vb, ERROR_SD);
if (va_power < 0.0) va_power = MINIMUM_POWER;
if (vb_power < 0.0) vb_power = MINIMUM_POWER;
power_ratio = va_power / vb_power;
minn = warcost_minimum(WARCOSTMAX, WCDISPAR);
warcostfactor = cal_warcost(va_power, vb_power, WARCOSTMAX);
lv = likelihood_of_victory(power_ratio, LVSIGMA);

// Cost
// ---------------------------------------------------------------------------
if (va_power > vb_power)
{
    a_warcost = a_power * (warcostfactor - minn);
b_warcost = b_power * (warcostfactor + minn);
    va_warcost = va_power * (warcostfactor - minn);
    vb_warcost = vb_power * (warcostfactor + minn);
}
else
{
    a_warcost = a_power * (warcostfactor + minn);
b_warcost = b_power * (warcostfactor - minn);
    va_warcost = va_power * (warcostfactor + minn);
    vb_warcost = vb_power * (warcostfactor - minn);
}
a_weight = a_power / va_power;
va_indem_win = (vb_power - vb_warcost) * REPAR;
a_indem_win = va_indem_win * a_weight;
if (vb.size() == 1) // if opposition is only 1 state
{
    if (vs[vb[0]].get_sc_size() == 1)
    {
        if (va.size() == 0) // if A is the only winner
            gains = b_power - b_warcost;
        else // Alliance is composed of 1+ states
        {
            if (is_a_leader)
                gains = b_power - b_warcost -
                        (1.0 - a_weight) * va_indem_win;
            else
                gains = a_indem_win;
        }
    }
else // Opponent state has 1+ cells
    {
va_cells_win = round((float)lv * (float)b_cells);
a_cells_win = round((float)va_cells_win * (float)a_weight);
gains = a_indem_win +
       ((float)a_cells_win / (float)b_cells) * vb_power) -
       vb_warcost - va_indem_win;
    }
else // if opposition consists of 1+ states
{
    if (vs[b].get_sc_size() == 1)
    {
        if (va.size() == 0) // if A is the only winner
            gains = b_power * (1.0 - warcostfactor) +
                    vb_power * (1.0 - warcostfactor) * REPAR;
        else // Alliance is composed of 1+ states
        {
            if (is_assumed_strongest(a, va, vs, ERROR_SD))
            {
                float temp1, temp2, temp3, temp4;
                temp1 = b_power * (1.0 - warcostfactor);
                temp2 = temp1 * (1.0 - a_weight) * REPAR;
                temp3 = temp1 - temp2;
                temp4 = vb_power * (1.0 - warcostfactor) *
                         REPAR * a_weight;
                gains = temp3 + temp4;
            }
            else // A is not the strongest
                gains = a_indem_win;
        }
    }
else // Opponent leader has 1+ cells
    {
va_cells_win = round(b_cells * lv);
a_cells_win = round(va_cells_win * a_weight);
gains = a_indem_win +
       ((float)a_cells_win / (float)b_cells) *
(1.0 - warcostfactor) * (1.0 - REPAR));


// Losses
if (is_a_leader)
{
    if (a_cells == 1)  losses = a_power - a_warcost;  // if A has 1+ cells
    else
    {
        a_indem_lose = (a_power - a_warcost) * REPAR;
        a_cells_lose = a_cells * (1.0 - lv);
        losses = a_indem_lose +
                  (((float)a_cells_lose / (float)a_cells) *
                   (a_power - a_warcost - a_indem_lose));
    }
}
else  // if A is not a leader
{
    losses = (a_power - a_warcost) * REPAR;
}

// Expected utility
expected_utility = lv * gains - ((1.0 - lv) * losses) - a_warcost;

// Return expected utility
return expected_utility;

// Setting the trade unit size according to two powers
float set_trade_unit(float a, float b)
{
    float trade_unit;
    float smaller;
    if (a > b) smaller = b;
    else    smaller = a;

    if (smaller < 0.00001)
        trade_unit = (float)0.0;
    if ((smaller >= 0.00001) && (smaller < 0.0001))
        trade_unit = (float)0.00001;
    if ((smaller >= 0.0001) && (smaller < 0.001))
        trade_unit = (float)0.0001;
    if ((smaller >= 0.001) && (smaller < 0.01))
        trade_unit = (float)0.001;
    if ((smaller >= 0.01) && (smaller < 0.1))
        trade_unit = (float)0.01;
    if ((smaller >= 0.1) && (smaller < 1.0))
        trade_unit = (float)0.1;
    if (smaller >= 1.0)
        trade_unit = (float)1.0;
}
return trade_unit;
}

// Estimate the state power with parameters
float estimate_state_power(float ura, float urb, float ma, float mb)
{
    float a_element = pow(ura, ma/(ma+mb));
    float b_element = pow(urb, mb/(ma+mb));
    return a_element * b_element;
}

// Expected utility of state A toward B for trade
// The size of trade unit is based on the size of two states' power
float eu_trade_state(vector<State> & vs, vector<Cell> & vc, int a, int b,
                      float ERROR_SD)
{
    // Variables declarations
    float expected_benefit = 0.0; // Return value
    float aura = vs[a].get_state_ura();
    float aurb = vs[a].get_state_urb();
    float bura = vs[b].get_state_ura();
    float burb = vs[b].get_state_urb();
    int an = vs[a].get_sc_size();
    int bn = vs[b].get_sc_size();
    float a_ma = vs[a].get_per_ma() * an;
    float a_mb = vs[a].get_per_mb() * an;
    float b_ma = vs[b].get_per_ma() * bn;
    float b_mb = vs[b].get_per_mb() * bn;
    float amrs = (aurb/a_mb) / (aura/a_ma);
    float bmrs = (burb/b_mb) / (bura/b_ma);

    // Set price and quantity
    float price = sqrt(amrs*bmrs);
    float aquantity = 0.0; // Amount of RA to exchange at one time
    float bquantity = 0.0; // Amount of RB to exchange at one time
    float a_power = estimate_state_power(aura, aurb, a_ma, a_mb) * (1.0 + nr(1.0, ERROR_SD));
    float b_power = estimate_state_power(bura, burb, b_ma, b_mb) * (1.0 + nr(1.0, ERROR_SD));
    float trade_unit = set_trade_unit(a_power, b_power);
    if (price >=1.0) // Exchanged market value = 1.0 * price
    {
        aquantity = 1.0 * trade_unit;
        bquantity = price * trade_unit;
    }
    else // Exchanged market value = 1.0
    {
        aquantity = (1.0/price) * trade_unit;
        bquantity = 1.0 * trade_unit;
    }

    // If amrs is bigger than bmrs (A sells urb, B sells ura)
if ((amrs >= bmrs) && (aurb >= trade_unit) && (bura >= trade_unit) && (aquantity >= trade_unit) && (bquantity >= trade_unit))
{
    do
    {
        // Calculate the difference of power before/after trade
        float before_power =
            estimate_state_power(aura, aurb, a_ma, a_mb)
            * (1.0 + nr(1.0, ERROR_SD));
        aurb = aurb - bquantity;
        burb = burb + bquantity;
        bura = bura - aquantity;
        aura = aura + aquantity;
        float after_power =
            estimate_state_power(aura, aurb, a_ma, a_mb)
            * (1.0 + nr(1.0, ERROR_SD));
        float difference = after_power - before_power;
        expected_benefit += difference;

        // Recalculate parameters
        amrs = (aurb/a_mb) / (aura/a_ma);
        bmrs = (burb/b_mb) / (bura/b_ma);
        price = sqrt(amrs*bmars);
        if (price >= 1.0)
        {
            aquantity = 1.0 * trade_unit;
            bquantity = price * trade_unit;
        }
        else
        {
            aquantity = (1.0/price) * trade_unit;
            bquantity = 1.0 * trade_unit;
        }
    }
    while ((amrs >= bmrs) && (aurb >= trade_unit) && (bura >= trade_unit) && (aquantity >= trade_unit) && (bquantity >= trade_unit));
}
else if ((amrs < bmrs) && (aura >= trade_unit) && (burb >= trade_unit) && (aquantity >= trade_unit) && (bquantity >= trade_unit))
{
    do
    {
        // Calculate the difference of power before/after trade
        float before_power =
            estimate_state_power(aura, aurb, a_ma, a_mb)
            * (1.0 + nr(1.0, ERROR_SD));
        aura = aura - aquantity;
        bura = bura + aquantity;
        burb = burb - bquantity;
        aurb = aurb + bquantity;
        float after_power =
            estimate_state_power(aura, aurb, a_ma, a_mb)
            * (1.0 + nr(1.0, ERROR_SD));
        float difference = after_power - before_power;
    }
}
expected_benefit += difference;
// Recalculate parameters
amrs = (aurb/a_mb) / (aura/a_ma);
bmrs = (burb/b_mb) / (bura/b_ma);
price = sqrt(amrs*bmrs);
if (price >= 1.0)
{
    aquantity = 1.0 * trade_unit;
bquantity = price * trade_unit;
}
else
{
    aquantity = (1.0/price) * trade_unit;
bquantity = 1.0 * trade_unit;
}
while ((amrs < bmrs) && (aura >= trade_unit)
    && (burb >= trade_unit) && (aqantity >= trade_unit)
    && (bquantity >= trade_unit))
{
    if (expected_benefit > 0.0) return expected_benefit;
else
    return 0.0;
}

// If no trade is expected
else
{
    expected_benefit = 0.0;
    return expected_benefit;
}

// Trade between two states
// a_export_ura: 0 = A exports urb; 1 = A exports ura; 2 = no trade

void trade(vector<State> & vs, vector<Cell> & vc, float ERROR_SD,
    int a, int b, int & real_trade, int & a_export_ura,
    float & a_export_ratio, float & b_export_ratio)
{
    // Variables declarations
    float aura = vs[a].get_state_ura();
    float aurb = vs[a].get_state_urb();
    float bura = vs[b].get_state_ura();
    float burb = vs[b].get_state_urb();
    int an = vs[a].get_sc_size();
    int bn = vs[b].get_sc_size();
    float a_ma = vs[a].get_per_ma() * an;
    float a_mb = vs[a].get_per_mb() * an;
    float b_ma = vs[b].get_per_ma() * bn;
    float b_mb = vs[b].get_per_mb() * bn;
    float amrs = (aurb/a_mb) / (aura/a_ma);
    float bmrs = (burb/b_mb) / (bura/b_ma);

    // Set price and quantity
    float price = sqrt(amrs*bmrs);
    float aquantity = 0.0; // Amount of RA to exchange at one time
float bquantity = 0.0;  // Amount of RB to exchange at one time
float a_power = estimate_state_power(aura, aurb, a_ma, a_mb);
float b_power = estimate_state_power(bura, burb, b_ma, b_mb);
float trade_unit = set_trade_unit(a_power, b_power);
if (price >= 1.0) // Exchanged market value = 1.0 * price
{
    aquantity = 1.0 * trade_unit;
bquantity = price * trade_unit;
}
else // Exchanged market value = 1.0
{
    aquantity = (1.0/price) * trade_unit;
bquantity = 1.0 * trade_unit;
}

// If amrs is bigger than bmrs (A sells urb, B sells ura)
// -----------------------------------------------------------------------------------------------
if ((amrs >= bmrs) && (aurb >= trade_unit) && (bura >= trade_unit) && (aquantity >= trade_unit) && (bquantity >= trade_unit))
{
do
    // Add the exchanged amount to each state
    aurb = aurb - bquantity;
burb = burb + bquantity;
bura = bura - aquantity;
aura = aura + aquantity;
    vs[a].reduce_state_urb(vc, bquantity);
    vs[b].add_state_urb(vc, bquantity);
    vs[b].reduce_state_ura(vc, aquantity);
    vs[a].add_state_ura(vc, aquantity);

    // Recalculate parameters
    amrs = (aurb/a_mb) / (aura/a_ma);
bmrs = (burb/b_mb) / (bura/b_ma);
    price = sqrt(amrs*bmrs);
    if (price >= 1.0)
    {
        aquantity = 1.0 * trade_unit;
bquantity = price * trade_unit;
    }
    else
    {
        aquantity = (1.0/price) * trade_unit;
bquantity = 1.0 * trade_unit;
    }
} while ((amrs >= bmrs) && (aurb >= trade_unit) && (bura >= trade_unit) && (aquantity >= trade_unit) && (bquantity >= trade_unit));
real_trade = 1;
a_export_ura = 0; // 0 = A exports urb; 1 = ura
}

// If bmrs is bigger than amrs
// -----------------------------------------------------------------------------------------------
else if ((amrs < bmrs) && (aura >= trade_unit) && (burb >= trade_unit) && (aquantity >= trade_unit) && (bquantity >= trade_unit))
{
do

282

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
{ 
    // Add the exchanged amount to each state
    aura = aura - aquantity;
bura = bura + aquantity;
burb = burb - bquantity;
aurb = aurb + bquantity;
    vs[a].reduce_state_ura(vc, aquantity);
    vs[b].add_state_ura(vc, aquantity);
    vs[b].reduce_state_urb(vc, bquantity);
    vs[a].add_state_urb(vc, bquantity);

    // Recalculate parameters
    amrs = (aurb/a_mb) / (aura/a_ma);
bmrs = (burb/b_mb) / (bura/b_ma);
    price = sqrt(amrs*bmrs);
    if (price >= 1.0)
    {
        aquantity = 1.0 * trade_unit;
bquantity = price * trade_unit;
    }
    else
    {
        aquantity = (1.0/price) * trade_unit;
bquantity = 1.0 * trade_unit;
    }
} while ((amrs < bmrs) && (aura >= trade_unit)
    && (burb >= trade_unit) && (aquantity >= trade_unit)
    && (bquantity >= trade_unit));

    real_trade = 1;
a_export_ura = 1;

    // util_configure_defensive_alliance()
    // util_configure_defensive_alliance (vector<State> & vs,
    // vector<Cell> & vc, int pinitiator, int ptarget, float ERROR_SD,
    // float WARCOSTMAX, float WCDISPAR, float REPAR, float LVSIGMA)
    
    vector<int> psally;
    vector<int> defensive_spv;
    for (int a = 0; a < vs[pinitiator].get_sns_size(); a++)
        if (vs[pinitiator].get_sns(a) != ptarget)
        {
            int state = vs[pinitiator].get_sns(a);
            float power = vs[state].get_state_power()
                * (1.0 + nr((float)0.0, (float)ERROR_SD));
            state_power sp;
            sp.state = state;
            sp.power = power;
    
    283

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
psally.push_back(sp);

// Sort the order of psally by power from the weakest to the strongest
sort(psally.begin(), psally.end(), by_state_power);
if (psally.size() == 0)
{
    defensive_spv.push_back(ptarget);
    return defensive_spv;
}

// ivv is the vector of binary numbers from zero to the number of psally
vector<vector<int>> ivv;
for (int i = 0; i < psally.size(); i++)
{
    vector<int> temp = binary(i);
    ivv.push_back(temp);
}

// svv is the vector of every combination of states (including target)
vector<vector<int>> svv;
for (int i = 0; i < ivv.size(); i++)
{
    vector<int> temp = ivv[i];
    for (int j = 0; j < temp.size(); j++)
    {
        vector<int> tempsv;
        if (temp[j] == 1)
        {
            tempsv.push_back(psally[ivv[i].size() - 1 - j].state);
            svv.push_back(tempsv);
        }
    }
}
if (svv.size() == 0)
{
    defensive_spv.push_back(ptarget);
    return defensive_spv;  // defensive_spv includes ptarg et
}

// Add the target to the defensive alliance configuration
for (int i = 0; i < svv.size(); i++)
    svv[i].push_back(ptarget);

// Choose the first coalition with positive EU
// -----------------------------------------------------------------------------------------------
float max_uv;
int sentinel = 0;  // if there is the first sv, then sentinel = 1
vector<int> vinitiator; vinitiator.push_back(pinitiator);
for (int i = 0; i < svv.size(); i++)
{
    vector<int> vtarget = svv[i];
    float temp_uv = exp_util_alliance(vs, vc, ERROR_SD, WARCOSTMAX,
        WCDISPAR, REPAR, LVSIGMA, ptarget, pinitiator,
        vtarget, vinitiator, true);
    if (temp_uv > 0.0)
    {
        defensive_spv = vtarget;
        max_uv = temp_uv;
    }
sentinel = 1;
break;
}

// Choose the max coalition with positive EU
if ((sentinel == 1) && (svv.size() > 1))
{
    for (i = 0; i < svv.size(); i++)
    {
        vector<int> vtarget = svv.at(i);
        float temp_uv = exp_util_alliance(vs, vc, ERROR_SD, WARCOSTMAX, WCDISPAR, REPAR, LVSIGMA, ptarget, pinitiator, vtarget, vinitiator, true);
        if (temp_uv > max_uv)
        {
            defensive_spv = vtarget;
            max_uv = temp_uv;
        }
    }
    return defensive_spv;  // defensive_spv includes ptarget
}

vector<int> util_build_defensive_alliance(vector<State> & vs, vector<Cell> & vc,
int pinitiator, int ptarget, vector<int> defensive_spv,
float ERROR_SD, float WARCOSTMAX, float WCDISPAR,
float REPAR, float LVSIGMA)
{
    vector<int> defensive_alliance;
    vector<int> temp_alliance;
    for (int i = 0; i < defensive_spv.size(); i++)
    {
        temp_alliance.push_back(defensive_spv.at(i));
    }
    defensive_alliance.push_back(ptarget);
    vector<int> vinitiator; vinitiator.push_back(pinitiator);
    for (int p = 0; p < temp_alliance.size(); p++) // Including the target
    {
        int temp = temp_alliance.at(p);
        if (temp == ptarget) continue;
        float temp_uv = exp_util_alliance(vs, vc, ERROR_SD, WARCOSTMAX, WCDISPAR, REPAR, LVSIGMA, temp, pinitiator,
temp_alliance, vinitiator, false);
        if (temp_uv > 0.0) // if utility > 0.0, join the alliance
        { defensive_alliance.push_back(temp); }
    }
    return defensive_alliance; // includes the ptarget
}

vector<int> util_configure_offensive_alliance(vector<State> & vs,
vector<Cell> & vc, int pinitiator, int ptarget,
vector<int> defensive_alliance, float ERROR_SD,
float WARCOSTMAX, float WCDISPAR, float REPAR, float LVSIGMA)
// psally is the list of all potential candidates for alliance
vector<state_power> psally;
vector<int> offensive_spv;
vector<int>::iterator iter;

// Find candidate states and insert its number to the vector
for (int a = 0; a < vs[ptarget].get_sns_size(); a++)
{
    int index = vs[ptarget].get_sns(a);
    iter = find(defensive_alliance.begin(),
                defensive_alliance.end(), index);
    if ((vs[index].get_id() != pinitiator) &&
        (iter == defensive_alliance.end()))
    {
        state_power sp;
        sp.state = index;
        sp.power = vs[index].get_state_power()
                   * (1.0 + nr((float)0.0, (float)ERROR_SD));
        psally.push_back(sp);
    }
}

// Sort the order of psally by their power from the weakest
sort(psally.begin(), psally.end(), by_state_power);
// If there is no candidates to ally
if (psally.size() == 0)
{
    offensive_spv.push_back(pinitiator);
    return offensive_spv;
}

// ivv is the vector of binary numbers from zero to the number of psally
vector<vector<int>> ivv;
for (int i = 0; i < psally.size(); i++)
{
    vector<int> temp;
    temp = binary(i);
    ivv.push_back(temp);
}

// svv is the vector of every combination of states (including initiator)
vector<vector<int>> svv;
for (i = 0; i < ivv.size(); i++)
{
    vector<int> temp = ivv[i];
    for (int j = 0; j < temp.size(); j++)
    {
        vector<int> tempsv;
        if (temp[j] == 1)
        {
            tempsv.push_back(psally[ivv[i].size() - 1 - j].state);
            svv.push_back(tempsv);
        }
    }
}

if (svv.size() == 0)
{

offensive_spv.push_back(pinitiator);
return offensive_spv;
}
// Add the target to offensive alliance configurations
for (i = 0; i < svv.size(); i++)
svv[i].push_back(pinitiator);

// --------------------------------------------------------------------------------------------
// Choose the first coalition with positive EU
// --------------------------------------------------------------------------------------------
float max_uv;
int sentinel = 0;  // if there is the first sv, then sentinel = 1
for (i = 0; i < svv.size(); i++)
{
    vector<int> vinitiator = svv.at(i);
    float temp_uv = exp_util_alliance(vs, vc, ERROR_SD, WARCOSTMAX,
                                     WCDISPAR, REPAR, LVSIGMA, pinitiator, ptarget,
                                     vinitiator, defensive_alliance, true);
    if (temp_uv > 0.0)
    {
        offensive_spv = vinitiator;
        max_uv = temp_uv;
        sentinel = 1;
        break;
    }
}
// --------------------------------------------------------------------------------------------
// Choose the max coalition with positive EU
// --------------------------------------------------------------------------------------------
if ((sentinel == 1) && (svv.size() > 1))
{
    for (i = 0; i < svv.size(); i++)
    {
        vector<int> vinitiator = svv.at(i);
        float temp_uv = exp_util_alliance(vs, vc, ERROR_SD,
                                           WARCOSTMAX, WCDISPAR, REPAR, LVSIGMA, pinitiator,
                                           ptarget, vinitiator, defensive_alliance, true);
        if (temp_uv > max_uv)
        {
            offensive_spv = vinitiator;
            max_uv = temp_uv;
        }
    }
}
return offensive_spv; // includes the initiator

// --------------------------------------------------------------------------------------------
// util_build_offensive_alliance()  
// --------------------------------------------------------------------------------------------
vector<int> util_build_offensive_alliance
(vector<State> & vs, vector<Cell> & vc, int pinitiator,
 int ptarget, vector<int> defensive_alliance,
 vector<int> offensive_spv, float ERROR_SD, float WARCOSTMAX,
 float WCDISPAR, float REPAR, float LVSIGMA)
{
    vector<int> offensive_alliance;
    vector<int> temp_alliance;
    for (int i = 0; i < offensive_spv.size(); i++)
        temp_alliance.push_back(offensive_spv.at(i));

    287

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
of fensive_alliance.push_back (pinitiator);
for (int p = 0; p < temp_alliance.size (); p++) // Including the target

(
int temp = temp_alliance.at(p);
if (temp == pinitiator) continue;
float temp_uv = exp_util_alliance (vs, vc, ERR0R_SD, WARCOSTMAX,
WCDISPAR, REPAR, LVSIGMA, temp, pinitiator,
temp_alliance, defensive_alliance, false);
if (temp_uv >0.0)
/ / i f utility > 0.0, join the alliance
offensive alliance.push_back(temp);

)
return offensive alliance;

// ============================================================================
// util_configure_enhanced_alliance ()

//

=========================================================================

vector<int> util_configure_enhanced_alliance (vector<State> & vs,
vector<Cell> & vc, int pinitiator, int ptarget,
vector<int> defensive_alliance, vector<int> offensive_alliance,
float ERROR_SD, float WARCOSTMAX, float WCDISPAR,
float REPAR, float LVSIGMA)

{
// psally is the list of all potential candidates for alliance
// the existing members of defensive_alliance will be added automatically
// Also, defensive_alliance already includes the target
vector<state_power> psally;
// The value to be returned; does not include the target state
vector<int> enhanced_spv;
vector<int>::iterator iterl, iter2;
// Find extra candidate states and insert its id to the vector
for (int a = 0; a < vs [pinitiator] .get_sns_size(); a++)

{
int index = vs[pinitiator].get_sns(a);
iterl = find(defensive_alliance.begin(),
defensive_alliance.end(), index) ;
iter2 = find(offensive_alliance.begin(),
offensive_alliance.end(), index) ;
if ((vs[index].get_id() != ptarget) &&
(iterl == defensive_alliance.end()) &&
(iter2 == offensive_alliance.end()))

{
state_power sp;
sp.state = index;
sp.power = vs[index].get_state_power()
* (1.0 + n r ((float)0.0, (float)ERROR_SD));
psally.push_back(sp);

)

)

if (psally.size() == 0)

{
enhanced_spv.clear () ;
return enhanced_spv;

// return the empty vector

>
// i w is the vector of binary numbers from zero to the number of psally
vector< vector<int> > i w ;
for (int i = 0; i < psally.size(); i++)

{
vector<int> temp;

288

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.


temp = binary(i);
ivv.push_back(temp);

// svv is the vector of every combination of states
vector<vector<int>> svv;
for (i = 0; i < ivv.size(); i++)
{
    vector<int> temp = ivv[i];
    for (int j = 0; j < temp.size(); j++)
    {
        vector<int> tempsv;
        if (temp[j] == 1)
        {
            tempsv.push_back(psally[i].size() - 1 - j].state);
            svv.push_back(tempsv);
        }
    }
}
if (svv.size() == 0)
{
    enhanced_spv.clear();
    return enhanced_spv; // return as an empty vector
}

float max_uv;
int sentinel = 0; // if there is the first sv, then sentinel = 1
// svv holds only the possible combinations of new candidates
for (i = 0; i < svv.size(); i++)
{
    vector<int> temp_vector_1 = svv.at(i); // only new candidates
    vector<int> temp_vector_2; // total combination holder
    temp_vector_2 = temp_vector_1; // add new candidates
    // then add the old defensive alliance members
    for (int j = 0; j < defensive_alliance.size(); j++)
    {temp_vector_2.push_back(defensive_alliance.at(j));
    float temp_uv = exp_util_alliance(vs, vc, ERROR_SD, WARCOSTMAX, WCDISPAR, REPAR, LVSIGMA, ptarget, pinitiator, temp_vector_2, offensive_alliance, true);
    if (temp_uv > 0.0)
    {
        enhanced_spv = temp_vector_1; // only new candidates
        max_uv = temp_uv;
        sentinel = 1;
        break;
    }
}

if ((sentinel == 1) && (svv.size() > 1))
{
    for (i = 0; i < svv.size(); i++)
    {
        289
    }
}
vector<int> temp_vector_3 = svv.at(i);
vector<int> temp_vector_4;
temp_vector_4 = temp_vector_3;
float temp_uv = exp_util_alliance(vs, vc, ERROR_SD, WARCOSTMAX, WCDISPAR, REPAR, LVSIGMA, ptarget, pinitiator, temp_vector_3, offensive_alliance, true);
if (temp_uv > max_uv)
{
    enhanced_spv = temp_vector_3;
    max_uv = temp_uv;
}

// enhanced_spv holds all the defensive members + new candidates
// The number of new candidates may be zero
return enhanced_spv;

}  // == = = = == = == == == == == == == == == == == == == == == == == == == == == == == == == == == == == =

vector<int> util_build_enhanced_alliance(vector<State> & vs, vector<Cell> & vc, int pinitiator, int ptarget, vector<int> defensive_alliance,
vector<int> offensive_alliance, vector<int> enhanced_spv, float ERROR_SD, float WARCOSTMAX, float WCDISPAR,
float REPAR, float LVSIGMA)
{
    // enhanced_alliance includes both the target and original
    // defensive_alliance members, plus new additional members
    vector<int> enhanced_alliance;
    for (int i = 0; i < defensive_alliance.size(); i++)
        enhanced_alliance.push_back(defensive_alliance.at(i));
    vector<int> temp_alliance;
    // Put the defensive_alliance members into temp_alliance
    for (i = 0; i < defensive_alliance.size(); i++)
        temp_alliance.push_back(defensive_alliance.at(i));
    // Then add the extra candidates
    for (i = 0; i < enhanced_spv.size(); i++)
        temp_alliance.push_back(enhanced_spv.at(i));
    vector<int>::iterator iter;
    for (int p = 0; p < temp_alliance.size(); p++)
    {
        int temp = temp_alliance.at(p);
        iter = find(defensive_alliance.begin(), defensive_alliance.end(), temp);
        // if the candidate is the target or any of original members
        if (iter != defensive_alliance.end()) continue;
        float temp_uv = exp_util_alliance(vs, vc, ERROR_SD, WARCOSTMAX, WCDISPAR, REPAR, LVSIGMA, temp, pinitiator, temp_alliance, offensive_alliance, false);
        if (temp_uv > 0.0) // if utility > 0.0, join the alliance
            enhanced_alliance.push_back(temp);
    }
    // Returns original defensive members + new bid accepters
    // the number of new bid accepters may be zero
    return enhanced_alliance;
}
Mathfunc.h
Function declarations for mathematical calculation

Some functions are replicated from Cusack and Stoll’s original
program - those functions are marked in comments
Written Aug. 1, 2000

#ifndef _MATHFUNC_H_
define _MATHFUNC_H_
#include "Param.h"

// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// signumO — from Cusack and Stoll
// Help function for the calculation of lv
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
float signum(float a, float b);

// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// c351() — from Cusack and Stoll
// Help function for the calculation of lv
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
float c351(float z );

// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// likelihood_of_victory() — from Cusack and Stoll
// The initiator’s likelihood of victory
// based on its relative power to the target (t)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
float likelihood_of_victory(float relpower, float LVSIGMA);

// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// Random number seed generation
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
void randomize();

// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// Generates an integer between lower and upper
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
int random_integer(int lower, int upper);

// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// Generates a float number between lower and upper
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
float random_float(float lower, float upper);

// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// Generates random numbers with uniform distribution between 0.0 and 1.0
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
float uniform_random();

// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// Generates random numbers with normal distribution
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
float nr(float mean, float sigma);
// Rounding a float number
int round(float number);

// Returns the maximum of two float numbers
float max_of_two(float number1, float number2);

// Returns the minimum of two float numbers
float min_of_two(float number1, float number2);

// Calculate dcon
float calculate_dcon(float a, float b);

#endif // _MATHFUNC_H_
// Function definitions for mathematical calculation

// Some functions are replicated from Cusack and Stoll's original
// program - those functions are marked in comments
// Written Aug. 18, 2000

#include <iostream>
#include <cmath>
#include <ctime>
#include "Mathfunc.h"

using namespace std;

// signumO — from Cusack and Stoll
// Help function for the calculation of lv
float signum(float a, float b)
{
    if (b >= 0.0)
    {
        if (a < 0.0)
            a = a * (-1.0);
    }
    else
    {
        if (a > 0.0)
            a = a * (-1.0);
    }
    return a;
}

// c351() — from Cusack and Stoll
// Help function for the calculation of lv
float c351(float z)
{
    float x, erf, pnom, qnom, reslt;
    x = fabs(z);
    erf = 1.0;
    if (x > 8.0)
        reslt = signum(erf, z);
    else
    {
        reslt = 0.0;
        if (x != 0.0)
        {
            pnom = ((((( 0.564130520296681 * x + 6.29498879129803 ) * x + 31.4993973347738 ) * x + 88.7286165525676 ) * x + 141.832105684723 ) * x + 114.091294767966;
            qnom = ((((( x + 11.1551294880233 ) * x + 31.4993973347738 ) * x + 6.29498879129803 ) * x + 0.564130520296681;
+ 56.3592284349702 ) * x
+ 162.636690952738 ) * x
+ 279.943263286653 ) * x
+ 270.570345847538 ) * x
+ 114.091294767967;

erf = 1.0 - pnom / qnom * exp (-x * x);
resit = signum(erf, z);
}
}
  
return resit;

// likelihood_of_victory() -- from Cusack and Stoll
// The initiator's likelihood of victory
// based on its relative power to the target (t)

float likelihood_of_victory(float relpower, float LVSIGMA)
{
  if (relpower == 0.0)
      return 0.0;
  return (float) (1.0/2.0
                   * (1.0 + (c351((1.0/sqrt(2.0))
                        * (log(relpower)/LVSIGMA))));
}

// Random number seed generation
void randomize()
{
  srand((int) time(NULL));
}

// Generates an integer between lower and upper
int random_integer(int lower, int upper)
{
  int k;
  float f;
  f = (float) rand() / ((float) RAND_MAX + 1);
  k = (int) (f * (upper - lower + 1));
  return (lower + k);
}

// Generates a float number between lower and upper
float random_float(float lower, float upper)
{
  float f;
  f = (float) rand() / ((float) RAND_MAX + 1.0);
  return (lower + f * (upper - lower));
}

// Generates random numbers with uniform distribution between 0.0 and 1.0
float uniform_random()
{ float f;
  f = (float) rand() / ((float) RAND_MAX + 1);
  return f;
}

// Generates random numbers with normal distribution
//======================================================================
float nr(float mean, float sigma)
{
  float sum = 0.0;
  for (int i = 0; i <= 12; i++)
    sum = sum + uniform_random();
  return (float) ((sum-6.0)*sigma + mean);
}

// Rounding a float number
//======================================================================
int round(float number)
{
  int truncnr = (int) number;
  float floatnr = (float) truncnr + (float) 0.5;
  if (number >= floatnr) return truncnr + 1;
  else return truncnr;
}

// Returns the maximum of two float numbers
//======================================================================
float max_of_two(float number1, float number2)
{
  if (number1 > number2) return number1;
  else return number2;
}

// Returns the minimum of two float numbers
//======================================================================
float min_of_two(float number1, float number2)
{
  if (number1 < number2) return number1;
  else return number2;
}

// Calculate dcon
//======================================================================
float calculate_dcon(float a, float b)
{
  float dcon = 0.0;
  float temp = a*a + b*b - 0.5;
  if (temp < 0.0)
    temp = 0.0;  // in case a == b with very small rounding error
  dcon = (float)sqrt(temp);
  return (float)dcon;
}
#ifndef _PREWAR_H_
define _PREWAR_H_
#include <vector>
#include <list>
#include "Cell.h"
#include "State.h"
#include "Param.h"
#include "Expected.h"
using namespace std;

struct state_power {
    int state;
    float power;
};

struct sl_power {
    list<State *> si;
    float power;
};

// Help function for the sorting of a state_power vector
bool by_state_power(state_power a, state_power b);

// binary(): Returns the number of whole combination of
// a vector of n elements (excluding empty combination) (i.e., 2**n - 1)
vector<int> binary(int n);

// choose_initiator(): Chooses an initiator from the world
int choose_initiator(vector<state_eu> s_positive_vseu,
float random_number, int & ptarget);

// configure_defensive_alliance()
// Configures the minimum requirements for the
defensive alliance and returns a list of possible candidates that meets the
// minimum winning coalition requirements
vector<int> configure_defensive_alliance(vector<State> & vs, int pinitiator,
int ptarget, float t_error_t, float t_error_i, float ERROR_SD);
// build_defensive_alliance()
// Each candidate of the candidates calculates
// the total power of the suggested coalition; Joins if it decides that the
// suggested coalition is more powerful than the initiator

vector<int> build_defensive_alliance (vector<State> & vs, int pinitiator,
    int ptarget, vector<int> defensive_spv, float ERROR_SD);

// configure_offensive_alliance()
// Initiator compares its power and the
// defensive alliance's power; If it's stronger than the defensive coalition,
// the war starts; it tries offensive coalition if it thinks that it is
// weaker than the defensive alliance.

vector<int> configure_offensive_alliance (vector<State> & vs, int pinitiator,
    int ptarget, float i_error_i, float i_error_t, float i_power_d,
    vector<int> defensive_spv, vector<int> defensive_alliance, float ERROR_SD);

// build_offensive_alliance()
// Each candidate in the offensive coalition
// compares the power of the defensive coalition and that of the proposed
// offensive coalition; If the latter looks like stronger, each joins the
// offensive alliance, otherwise it gives up.

vector<int> build_offensive_alliance (vector<State> & vs, int pinitiator,
    int ptarget, vector<int> defensive_alliance,
    vector<int> offensive_spv, float ERROR_SD);

// configure_enhanced_alliance()
// After any offensive alliance is formed,
// target configures whether any addition of defensive alliance is available.

vector<int> configure_enhanced_alliance (vector<State> & vs, int pinitiator,
    int ptarget, vector<int> defensive_spv,
    vector<int> defensive_alliance, vector<int> offensive_alliance,
    float t_power_d, float difference, float ERROR_SD);

// build_enhanced_alliance()
// Each candidates of the enhanced defensive
// coalition compares the power of offensive alliance and that of the newly
// proposed enhanced defensive alliance; if the latter looks like stronger,
// it joins it, otherwise, it does not.

vector<int> build_enhanced_alliance (vector<State> & vs, int pinitiator,
    int ptarget, vector<int> offensive_alliance,
    vector<int> defensive_alliance, vector<int> enhanced_spv,
    float ERROR_SD);

#ifdef _PREWAR_H_

#endif _PREWAR_H_
/* Prewar.cpp
 Function definitions for war initiation and alliance formation

 Programmed May 31, 2000

 */

#include <iostream>
#include <iomanip>
#include <fstream>
#include <stdlib.h>
#include <algorithm>
#include <ctime>
#include <cmath>
#include "Prewar.h"
#include "Mathfunc.h"

using namespace std;

/*
 Help function for the sorting of a state_power vector
 */
bool by_state_power(state_power a, state_power b)
{
    if (a.power < b.power) return true;
    else return false;
}

/*
 binary(): Returns the number of whole combination of
 a list of n elements (excluding empty combination) (i.e., 2**n - 1)
 */
vector<int> binary(int n)
{
    int quotient = n;
    vector<int> iv;
    if (quotient == 0 )
    {  
        iv.push_back(0) ;
        return iv;
    }
    while(quotient > 0 )
    {
        iv.push_back(quotient %  2) ;
        quotient = quotient /  2;
    }
    reverse(iv.begin(), iv.end());
    return iv;
}

/*
 choose_initiator(): Chooses an initiator from the world
 */
int choose_initiator(vector<state_eu> & positive_vseu,
        float random_number, int & ptarget)
{
    float accumulated = 0.0;
    int index = 0;
if (positive_vseu.size() == 0)
{
    cerr << "Error: Size of positive_vseu = 0";
    exit(1);
}

while (accumulated <= random_number)
{
    //accumulated += positive_vseu[index].weighted_eu;
    accumulated += positive_vseu[index].relative_real_power;
    index++;
}

ptarget = positive_vseu[index-1].target;
return positive_vseu[index-1].id;

// ========================================================

// configure_defensive_alliance()

// Configures the minimum requirements for the defensive alliance and returns a list of possible candidates that meets the minimum winning coalition requirements

vector<int> configure_defensive_alliance (vector<State> & vs, int pinitiator, int ptarget, float t_error_t, float t_error_i, float ERROR_SD)
{
    // [ Minimum Winning Coalition Requirements ]
    // Three conditions should be met for building of defensive alliance:
    // (1) All possible coalitions of states contiguous to opponent assayed.
    // (2) The combined power of each such potential coalition is estimated
    // (3) Alliance membership bids are directed toward those states in group
    //   - which is contiguous to the opponent,
    //   - which has more combined power, and yet
    //   - which is the least powerful of all such proto-coalitions

    vector<state_power> psally;
    vector<int> defensive_spv;

    for (int a = 0; a < vs[pinitiator].get_sns_size(); a++)
    {
        if (vs[pinitiator].get_sns(a) != ptarget)
        {
            int state = vs[pinitiator].get_sns(a);
            float power = vs[state].get_state_power()
                          * (1.0 + nr((float)0.0, (float)ERROR_SD));
            state_power sp;
            sp.state = state;
            sp.power = power;
            psally.push_back(sp);
        }
    }

    sort(psally.begin(), psally.end(), by_state_power);

    // Sort the order of psally by their power from the weakest
    // If there is no candidates to ally, return an empty vector
if (psally.size() == 0)    return defensive_spv;
float t_power_t = vs[ptarget].get_state_power() * (1.0 + t_error_t);
float t_power_i = vs[pinitiator].get_state_power() * (1.0 + t_error_i);

// ivv is the vector of binary numbers from zero to the number of psally
vector<vector<int> > ivv;
for (int i = 0; i < psally.size(); i++)
{
    vector<int> temp;
    temp = binary(i);
    ivv.push_back(temp);
}

// svv is the vector of every combination of states (not target)
// pvv is the vector of the power of each state in combinations
// svv and pvv match in their order and numbers
vector<vector<int> > svv;
vector<vector<float> > pvv;
for (i = 0; i < ivv.size(); i++)
{
    vector<int> temp = ivv[i];
    for (int j = 0; j < temp.size(); j++)
    {
        vector<int> tempsv;
        vector<float> temppv;
        if (temp[j] == 1)
        {
            tempsv.push_back(psally[i].size() - 1 - j).state);
            temppv.push_back(psally[i].size() - 1 - j).power);
            svv.push_back(tempsv);
            pvv.push_back(temppv);
        }
    }
}
if (svv.size() == 0)
    return defensive_spv; // return an empty vector

// Choose the coalitions from the svv (and pvv) that meets requirements
// mentioned above: (1), (2), and (3)
vector<vector<int> > oversv;
vector<vector<float> > overpv;
for (i = 0; i < svv.size(); i++)
{
    vector<int> tempsv = svv.at(i);
    vector<float> temppv = pvv.at(i);
    float temppv_total = 0.0;
    for (int k = 0; k < temppv.size(); k++)
        temppv_total += temppv.at(k);
    if (temppv_total + t_power_t > t_power_i)
    {
        oversv.push_back(tempsv);
        overpv.push_back(temppv);
    }
}
if (oversv.size() == 0)
    return defensive_spv; // return an empty vector
/ Choose the minimum power coalition

for (i = 0; i < oversv.size(); i++)
{
    float minimum_power;
    if (i == 0)
    {
        defensive_spv = oversv.at(0);
        for (int k = 0; k < overpv.at(0).size(); k++)
            minimum_power += overpv.at(0).at(k);
    }
    else
    {
        float current_power = 0.0;
        for (int m = 0; m < overpv.at(i).size(); m++)
            current_power += overpv.at(i).at(m);
        if (current_power < minimum_power)
        {
            defensive_spv = oversv.at(i);
            minimum_power = current_power;
        }
    }
} // defensive_spv does not include the target state
return defensive_spv;

// build_defensive_alliance()
// Each candidate of the candidates calculates
// the total power of the suggested coalition; Joins if it decides that the
// suggested coalition is more powerful than the initiator

vector<int> build_defensive_alliance (vector<State> & vs, int pinitiator,
    int ptarget, vector<int> defensive_spv, float ERROR_SD)
{
    // Return value; does not include the target state
    vector<int> defensive_alliance;
    for (int p = 0; p < defensive_spv.size(); p++) // Excluding the target
    {
        float d_power_i = vs[pinitiator].get_state_power()
            * (1.0 + nr((float)0.0, (float)ERROR_SD));
        float d_power_d = vs[ptarget].get_state_power()
            * (1.0 + nr((float)0.0, (float)ERROR_SD));
        for(int m = 0; m < defensive_spv.size(); m++)
            d_power_d += vs[defensive_spv[m]].get_state_power()
                * (1.0 + nr((float)0.0, (float)ERROR_SD));
        if (d_power_d > d_power_i)
            defensive_alliance.push_back(defensive_spv[p]);
    }
    // The list of alliance candidates who accept the bids
    // defensive_alliance does not include the target state
    // defensive_alliance may be less than the original configuration
    return defensive_alliance;
}
configure_offensive_alliance( )

// Initiator compares its power and the defensive alliance's power; If it's stronger than the defensive coalition, the war starts; it tries offensive coalition if it thinks that it is weaker than the defensive alliance.

vector<int> configure_offensive_alliance (vector<State> & vs, int pinitiator,
                                       int ptarget, float i_error_i, float i_error_t, float i_power_d,
                                       vector<int> defensive_alliance, float ERROR_SD)
{
    // psally is the list of all potential candidates for alliance
    vector<state_power> psally;
    // The value to be returned; does not include the initiator state
    vector<int> offensive_spv;
    vector<int>::iterator iter;
    // Now find the candidates and insert its number to the vector
    for (int a = 0; a < vs[ptarget].get_sns_size(); a++)
    {
        int index = vs[ptarget].get_sns(a);
        iter = find(defensive_alliance.begin(),
                     defensive_alliance.end(), index);
        if ((vs[index].get_id() != pinitiator) &&
            (iter == defensive_alliance.end()))
        {
            state_power sp;
            sp.state = index;
            sp.power = vs[index].get_state_power() * (1.0 + nr((float)0.0, (float)ERROR_SD));
            psally.push_back(sp);
        }
    }

    // Sort the order of psally by their power from the weakest
    sort(psally.begin(), psally.end(), by_state_power);
    // If there is no candidates to ally, return an empty vector
    if (psally.size() == 0) return offensive_spv;
    float i_power_i = vs[pinitiator].get_state_power() * (1.0 + i_error_i);
    // ivv is the vector of binary numbers from zero to the number of psally
    vector<vector<int> > ivv;
    for (int i = 0; i < psally.size(); i++)
    {
        vector<int> temp;
        temp = binary(i);
        ivv.push_back(temp);
    }

    // svv is the vector of every combination of states (not initiator)
    // pvv is the vector of the power of each state in combinations
    // svv and pvv match in their order and numbers
    vector<vector<int> > svv;
    vector<vector<float> > pvv;
    for (i = 0; i < ivv.size(); i++)
    {
        vector<int> temp = ivv[i];

    }
for (int j = 0; j < temp.size(); j++)
{
    vector<int> tempsv;
    vector<float> temppv;
    if (temp[j] == 1)
    {
        tempsv.push_back(psally[iw[i].size() - 1 - j].state);
        temppv.push_back(psally[iw[i].size() - 1 - j].power);
        sw.push_back(tempsv);
        pw.push_back(temppv);
    }
}
if (svv.size() == 0)
    return offensive_spv; // return an empty vector

// Choose the coalitions from the svv (and pvv) that meets requirements
// mentioned above: (1), (2), and (3)
// The least powerful coalitions and their power over the initiator's
vector<vector<int>> oversv;
vector<vector<float>> overpv;
for (i = 0; i < sw.size(); i++)
{
    vector<int> tempsv = sw.at(i);
    vector<float> temppv = pw.at(i);
    float temppv_total = 0.0;
    for (int k = 0; k < temppv.size(); k++)
        temppv_total += temppv.at(k);
    if (temppv_total + i_power_i > i_power_d)
    {
        oversv.push_back(tempsv);
        overpv.push_back(temppv);
    }
}
if (oversv.size() == 0)
    return offensive_spv; // return an empty vector

// Choose the minimum power coalition
for (i = 0; i < oversv.size(); i++)
{
    float minimum_power;
    if (i == 0)
    {
        offensive_spv = oversv.at(0);
        for (int k = 0; k < overpv.at(0).size(); k++)
            minimum_power += overpv.at(0).at(k);
    }
    else
    {
        float current_power = 0.0;
        for (int m = 0; m < overpv.at(i).size(); m++)
            current_power += overpv.at(i).at(m);
        if (current_power < minimum_power)
        {
            offensive_spv = oversv.at(i);
            minimum_power = current_power;
        }
    }
}
// offensive_spv does not include the initiator state
return offensive_spv;

// build_offensive_alliance()
// Each candidate in the offensive coalition
// compares the power of the defensive alliance and that of the proposed
// offensive alliance; If the latter looks like stronger, each joins the
// offensive alliance, otherwise it gives up.

vector<int> build_offensive_alliance(vector<State> & vs, int pinitiator,
int ptarget, vector<int> defensive_alliance,
vector<int> offensive_spv, float ERROR_SD)
{
    // Return value; does not include the initiator state
    vector<int> offensive_alliance;
    for (int p = 0; p < offensive_spv.size(); p++) // Except the initiator
    {
        // spv[p]'s estimation of the defensive alliance power
        float o_power_d = vs[ptarget].get_state_power()
        * (1.0 + nr((float)0.0, (float)ERROR_SD));
        for (int m = 0; m < defensive_alliance.size(); m++)
            o_power_d += vs[defensive_alliance[m]].get_state_power()
            * (1.0 + nr((float)0.0, (float)ERROR_SD));
        // spv[p]'s estimation of the offensive power
        float o_power_o = vs[pinitiator].get_state_power()
        * (1.0 + nr((float)0.0, (float)ERROR_SD));
        for (int n = 0; n < offensive_spv.size(); n++)
            o_power_o += vs[offensive_spv[n]].get_state_power()
            * (1.0 + nr((float)0.0, (float)ERROR_SD));
        if (o_power_o > o_power_d)
            offensive_alliance.push_back(offensive_spv[p]);
    }
    // The list of alliance candidates who accept the bids
    // offensive_alliance does not include the initiator state
    // offensive_alliance may be less than the original configuration
    return offensive_alliance;
}

// configure_enhanced_alliance()
// After any offensive alliance is formed,
// target configures whether any addition of defensive alliance is available.

vector<int> configure_enhanced_alliance(vector<State> & vs, int pinitiator,
int ptarget, vector<int> defensive_spv,
vector<int> defensive_alliance, vector<int> offensive_alliance,
float t_power_d, float difference, float ERROR_SD)
{
    // psally is the list of potential states for alliance
    // More candidates to be added
    vector<int> psally;
    // The value to be returned; does not include the target state
    vector<int> enhanced_spv;
    vector<int>::iterator iter1, iter2;
    // Now find extra candidates and insert its id to the vector
    for (int a = 0; a < vs[pinitiator].get_sns_size(); a++)
int index = vs[pinitiator].get_sns(a);
iter1 = find(defensive_alliance.begin(),
defensive_alliance.end(), index);
iter2 = find(offensive_alliance.begin(),
offensive_alliance.end(), index);
if ((vs[index].get_id() != ptarget) &&
(iter1 == defensive_alliance.end()) &&
(iter2 == offensive_alliance.end()))
{
    state_power sp;
    sp.state = index;
    sp.power = vs[index].get_state_power()
        * (1.0 + nr((float)0.0, (float)ERROR_SD));
    psally.push_back(sp);
}

// Sort the order of psally by their power from the weakest
sort(psally.begin(), psally.end(), by_state_power);
// If there is no candidates to ally, return an empty vector
if (psally.size() == 0) return enhanced_spv;
// t_power_d is passed from outside as a parameter
// difference is passed from outside as a parameter
float t_power_o = t_power_d + difference;
// ivv is the vector of binary numbers from zero to the number of psally
vector<vector<int>> iw;
for (int i = 0; i < psally.size(); i++)
{
    vector<int> temp;
    temp = binary(i);
    ivv.push_back(temp);
}

// svv is the vector of every combination of states (not target)
// pvv is the vector of the power of each state in combinations
// svv and pvv match in their order and numbers
// --------------------------------------------------------------------------------------------
vector<vector<int>> svv;
vector<vector<float>> pvv;
for (i = 0; i < ivv.size(); i++)
{
    vector<int> temp = ivv[i];
    for (int j = 0; j < temp.size(); j++)
    {
        vector<int> temp1;
        vector<float> temp2;
        for (int k = 0; k < temp.size(); k++)
        {
            temp1.push_back(temp[k]);
            temp2.push_back(temp[k].power);
        }
        svv.push_back(temp1);
        pvv.push_back(temp2);
    }
}
if (svv.size() == 0)
    return enhanced_spv; // return an empty vector

305
// Choose the coalitions from the svv (and pvv) that meets requirements mentioned above: (1), (2), and (3)
// The least powerful coalitions and their power over the initiator's
vector<vector<int>> oversv;
vector<vector<float>> overpv;
for (i = 0; i < sw.size(); i++)
{
    vector<int> tempsv = sw.at(i);
    vector<float> temppv = pvv.at(i);
    float temppv_total = 0.0;
    for (int k = 0; k < temppv.size(); k++)
        temppv_total += temppv.at(k);
    if (temppv_total + t_power_d > t_power_o)
    {
        oversv.push_back(tempsv);
        overpv.push_back(temppv);
    }
}
if (oversv.size() == 0)
    return enhanced_spv; // return an empty vector

// Choose the minimum power coalition
for (i = 0; i < oversv.size(); i++)
{
    float minimum_power;
    if (i == 0)
    {
        enhanced_spv = oversv.at(0);
        for (int k = 0; k < overpv.at(0).size(); k++)
            minimum_power += overpv.at(0).at(k);
    }
    else
    {
        float current_power = 0.0;
        for (int m = 0; m < overpv.at(i).size(); m++)
            current_power += overpv.at(i).at(m);
        if (current_power < minimum_power)
        {
            enhanced_spv = oversv.at(i);
            minimum_power = current_power;
        }
    }
}
// enhanced_spv does not include the target state
// nor any of the defensive alliance members
return enhanced_spv;

// build_enhanced_alliance()
// Each candidates of the enhanced defensive
// coalition compares the power of offensive alliance and that of the newly
// proposed enhanced defensive alliance; if the latter looks like stronger,
// it joins it, otherwise, it does not.
//
vector<int> build_enhanced_alliance (vector<State> & vs, int pinitiator,
int ptarget, vector<int> offensive_alliance,
vector<int> defensive_alliance, vector<int> enhanced_spv,
float ERROR_SD)
{
    // Returning value; enhanced_alliance does not include
    // the target state nor any of the original defensive alliance's members
    vector<int> enhanced_alliance;
    for(int q = 0; q < enhanced_spv.size(); q++)
    {
        // spv[q]'s estimation of existing defensive alliance's power
        float d_power_d = vs[ptarget].get_state_power()
            * (1.0 + nr((float)0.0, (float)ERROR_SD));
        for (int m = 0; m < defensive_alliance.size(); m++)
            d_power_d += vs[defensive_alliance[m]].get_state_power()
                * (1.0 + nr((float)0.0, (float)ERROR_SD));
        // spv[q]'s estimation of newly added candidates' power
        for(int n = 0; n < enhanced_spv.size(); n++)
            d_power_d += vs[enhanced_spv[n]].get_state_power()
                * (1.0 + nr((float)0.0, (float)ERROR_SD));
        // spv[q]'s estimation of offensive alliance
        float d_power_o = vs[pinitiator].get_state_power()
            * (1.0 + nr((float)0.0, (float)ERROR_SD));
        for (int s = 0; s < offensive_alliance.size(); s++)
            d_power_o += vs[offensive_alliance[s]].get_state_power()
                * (1.0 + nr((float)0.0, (float)ERROR_SD));

        // Add the state which accepted the bid
        // upon its own estimation of the two alliances' power
        if (d_power_d > d_power_o)
            enhanced_alliance.push_back(enhanced_spv.at(q));
    }
    // The list of alliance candidates who accept the bids
    // enhanced_alliance does not include the target state
    // nor any of the original defensive alliance's members
    // enhanced_alliance may be less than enhanced_spv in its size
    return enhanced_alliance;
}
# Prints.h

# Prints information to designated files

```c
#ifndef _PRINTS_H_
#define _PRINTS_H_

#include <iostream>
#include <iomanip>
#include <fstream>
#include <vector>
#include "Cell.h"
#include "State.h"

using namespace std;

// Print out the initial interactive screen
void print_screen(string & filename0, string & filename1, string & filename2,
                  string & filename3, int trade_on, char response);

// Print out the basic parameters
void print_parameters(ostreams fout, int run, int style, float POW_SD,
                      float ERROR_SD, float LVSIGMA, float WARCOSTMAX,
                      float WCDISPAR, float REPAR, float META_SD);

// Print out the number of run and round
void print_run_round(ostreams fout, int run, int round);

// Print out the polarity value, major state number, minor state number
void print_polarity(ostreams fout, int system_polarity, int major_size,
                    int minor_size, int total_size);

// Print out the expected utility of trade for initiator and target
void print_eu(ostreams fout, int i_major, int t_major, float i_power,
              float t_power, float bigger, float p_ratio, float dcon,
              float i_eu, float t_eu, float low_eu_ratio, float high_eu_ratio,
              float str_int, int pinitiator_wot);

// Print out the variable values for real trade or war happening
// (0 = not really happened; 1 = really happened)
void print_real(ostreams fout, int real_trade, int real_war);

// Print out war size value briefly
```

308

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
void print_war_size_brief(ostreams fout, int war_size);

// Print out the information about initiator and target
void print_screen_1(ostreams cout, int run, int round, int number_of_states);

// Print out the information about who wins
void print_screen_2(int war_or_trade, int index);

// Print out the initial setup information to a log file
void print_initial(ostreams fout, int run, int round, int number_of_states,
    vector<State> & s, vector<Cell> & c);

// Print out the information about initiator and target
void print_it(ostreams fout, int run, int round, int number_of_states,
    vector<State> & vs, vector<Cell> & vc, int pinitiator, int ptarget);

// Print out no war or trade event information to a log file
void print_no_event(ostreams fout, int run, int round, int number_of_states);

// Print out no war information to a log file
void print_no_war(ostreams fout, int run, int round, int number_of_states);

// Print out trade information to a log file
void print_trade(ostreams fout, int run, int round, int number_of_states);

// Print out trade information to a log file (details)
// ura_or_urb: 0 (A exports ura) 1 (A exports urb) 2 (No trade)
void print_trade_details(ostreams fout, vector<State> & vs, vector<Cell> & vc,
    int a, int b, float unit, float price, int ura_or_urb);

// Print out alliance configuration (for each run)
void print_alliance(ostreams fout, int run, int round, int number_of_states,
    vector<State> & vs, vector<int> offensive_part, float opower, int
    pinitiator, vector<int> defensive_part, float dpower, int ptarget);

// Print out lt, lv and rnd
void print_lvt(ostreams fout, int run, int round, int number_of_states,
    float lv, float lt, float rnd);

// Print out who wins
309
// Index = 1: initiator wins
// Index = 2: target wins
// Index = 3: tied
// ---------------------------------------------------------------
void print_who_wins(ostream & fout, int index);

// Print out cell claim vector
// ---------------------------------------------------------------
void print_cell_claim(ostream & fout, int cell_loss, vector<State> & vs,
        vector<int> victor_list, vector<float> victor_power,
        vector<float> victor_power_ratio,
        vector<int> precell_claim, vector<int> intcell_claim);

// Print out cell vector elements
// ---------------------------------------------------------------
void print_cell_vector(ostream & fout, vector<Cell> & vc,
        vector<int> cell_vector, string s);

// Print out cell transfer
// ---------------------------------------------------------------
void print_transfer(ostream & fout, vector<State> & vs, vector<Cell> & vc,
        int pcell, int ploser, int claimant);

// Print out the generation of a new state after cell transfer
// ---------------------------------------------------------------
void print_new_state(ostream & fout, vector<State> & vs, vector<Cell> & vc,
        int ploser, int pnew, int pcell);

// Print out the generation of a new state after a vector of cells transfer
// ---------------------------------------------------------------
void print_new_state(ostream & fout, State * ploser, State * ps,
        vector<Cell *> vs);

// Print out the size of war
// ---------------------------------------------------------------
void print_war_size(ostream & fout, int run, int round, int osize, int dsize,
        float ipower, float tpower, float opower, float dpower,
        float totalpower, int warcells);

// Print out the result of the simulation (dependent variables)
// ---------------------------------------------------------------
void print_result(ostream & fout, int new_state_number, int trade_freq,
        int no_war_freq, int war_freq, int target_wins,
        int initiator_wins, int real_round, int final_state_number);

// Print out the total world power
// ---------------------------------------------------------------
void print_world_total_power(ostream & fout, int run, int round,
        int number_of_states, float world_total_power);

#endif // PRINTS_H_
#include <string>
#include <windows.h>
#include "Prints.h"

void print_screen(string & filename0, string & filename1, string & filename2,
                 string & filename3, int & trade_on, char & response)
{
    string filename;

    HANDLE hConsole;
    WORD wAttributesOld;
    CONSOLE_SCREEN_BUFFER_INFO csbi;

    // Open the current console input buffer
    if ((hConsole = CreateFile ("CONOUTS", GENERIC_WRITE | GENERIC_READ,
                                 FILE_SHARE_READ | FILE_SHARE_WRITE,
                                 0L, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0L))
        == (HANDLE) -1)
    {
        cout << endl << "Error: Unable to open console." << endl;
        exit(1);
    }

    // Get and save information on the console screen buffer
    GetConsoleScreenBufferInfo(hConsole, &csbi);
    wAttributesOld = csbi.wAttributes;

    // Show the title and version information
    SetConsoleTextAttribute (hConsole, BACKGROUND_GREEN |
                             BACKGROUND_INTENSITY);
    cout << setw(80) << " " << endl;
    // green line
// Return back to the normal screen, then change to bright white text
SetConsoleTextAttribute(hConsole, wAttributesOld);
SetConsoleTextAttribute(hConsole, FOREGROUND_RED | FOREGROUND_GREEN | FOREGROUND_INTENSITY);
cout « setw(37) « " " << program_info_1 « endl;
cout « setw(18) « " " << program_info_2 « endl « endl;
SetConsoleTextAttribute(hConsole, FOREGROUND_RED | FOREGROUND_GREEN | FOREGROUND_BLUE | FOREGROUND_INTENSITY);
cout « setw(25) « " " << program_info_3;
SetConsoleTextAttribute(hConsole, FOREGROUND_RED | FOREGROUND_GREEN | FOREGROUND_INTENSITY);
cout « setw(80) « " " << endl;
SetConsoleTextAttribute(hConsole, BACKGROUND_GREEN | BACKGROUND_INTENSITY);
SetConsoleTextAttribute(hConsole, wAttributesOld);
cout « emptyline_10 « "FEATURES OF VERSION ";
SetConsoleTextAttribute(hConsole, FOREGROUND_RED | FOREGROUND_GREEN | FOREGROUND_INTENSITY);
cout « program_info_4;
SetConsoleTextAttribute(hConsole, wAttributesOld);
cout « ";
SetConsoleTextAttribute(hConsole, BACKGROUND_GREEN | BACKGROUND_INTENSITY);
cout « emptyline_10 « " " << "1. Major revision to include Cusack and Stoll's framework." « endl;
cout « emptyline_10 « " " << "2. Trade on/off module included."
<< endl « endl;

// Ask to type the output filename and initialize file
// ---------------------------------------------------------------------------
cout « emptyline_10 « "Type a filename " << "(extension ".txt" will be added): ";
cin « filename;
cout « emptyline_10 « "Turn on trade module? (Y/N): ";
char t;
string extension;
cin >> t;
while (!((t == 'y')|| (t == 'n')||(t == 'Y')||(t == 'N')))
{
cout << emptyline_10 << "Please enter "Y" or "N": ";
cin >> t;
}
if ((t == 'y')||(t == 'Y'))
{
    trade_on = 1;
    extension = "_on.txt";
}
else if ((t == 'n')||(t == 'N'))
{
    trade_on = 0;
}
extension = "_off.txt";

else
{
cerr << "Error in processing trade turn-on option." << endl;
cerr << "Please check the source code." << endl;
exit(1); // Exit to the system
}

// File initialization
// --------------------------------------------------------------------------------------------
filename0 = filename + "_log" + extension;
filename1 = filename + "_data" + extension;
filename2 = filename + "_int" + extension;
filename3 = filename + "_temp" + extension;

// Confirm the output filenames
// --------------------------------------------------------------------------------------------
SetConsoleTextAttribute (hConsole, FOREGROUND_RED | FOREGROUND_GREEN | FOREGROUND_INTENSITY);
cout << emptyline_10 << stringline_60 << endl;
SetConsoleTextAttribute(hConsole, wAttributesOld);
cout << emptyline_10 << emptyline_10 << "Basic data file: ";
SetConsoleTextAttribute(hConsole, FOREGROUND_RED | FOREGROUND_GREEN | FOREGROUND_INTENSITY);
cout << filename1 << endl;
SetConsoleTextAttribute(hConsole, wAttributesOld);
cout << emptyline_10 << emptyline_10 << "Interdependence data file: ";
SetConsoleTextAttribute(hConsole, FOREGROUND_RED | FOREGROUND_GREEN | FOREGROUND_INTENSITY);
cout << filename2 << endl;
SetConsoleTextAttribute(hConsole, wAttributesOld);
cout << emptyline_10 << emptyline_10 << "Temp file: ";
SetConsoleTextAttribute(hConsole, FOREGROUND_RED | FOREGROUND_GREEN | FOREGROUND_INTENSITY);
cout << filename3 << endl;
SetConsoleTextAttribute(hConsole, wAttributesOld);
cout << emptyline_10 << emptyline_10 << "Checkup log file: ";
SetConsoleTextAttribute(hConsole, FOREGROUND_RED | FOREGROUND_GREEN | FOREGROUND_INTENSITY);
cout << filename0 << endl;

// Ask to continue the process or not
// --------------------------------------------------------------------------------------------
SetConsoleTextAttribute(hConsole, FOREGROUND_RED | FOREGROUND_GREEN | FOREGROUND_INTENSITY);
cout << emptyline_10 << stringline_60 << endl;
SetConsoleTextAttribute(hConsole, wAttributesOld);
char response;
cout << emptyline_10 << "Press ";
SetConsoleTextAttribute(hConsole, FOREGROUND_RED | FOREGROUND_GREEN | FOREGROUND_INTENSITY);
cout << "Y";
SetConsoleTextAttribute(hConsole, wAttributesOld);
cout << " if you want to start, or ";
SetConsoleTextAttribute(hConsole, FOREGROUND_RED | FOREGROUND_GREEN | FOREGROUND_INTENSITY);
cout << "N";

313
SetConsoleTextAttribute(hConsole, wAttributesOld);
cout << " if you want to exit: ";
SetConsoleTextAttribute(hConsole, FOREGROUND_RED | FOREGROUND_GREEN | FOREGROUND_INTENSITY);

cin >> response;
SetConsoleTextAttribute(hConsole, wAttributesOld);

while ((response != 'Y') && (response != 'y')
   && (response != 'N') && (response != 'n'))
{
    cout << "Wrong input: please re-enter your response (Y or N):"
        << emptyline_10
        << "Wrong input: please re-enter your response (Y or N):"
        << emptyline_10
    cin >> response;
    SetConsoleTextAttribute(hConsole, wAttributesOld);
}

if ((response == 'N') || (response = 'n')) exit(1);

// Print out the basic parameters
void print_parameters(ostream& fout, int run, int style, float POW_SD, float ERROR_SD, float LVSIGMA, float WARCOSTMAX, float WCDISPAR, float REPAR, float META_SD)
{
    if (style == 1)
    {
        fout << setw(5) << right << run << "\t";
        fout << setw(8) << setprecision(4) << fixed << right
            << POW_SD << "\t";
        fout << setw(8) << setprecision(4) << fixed << right
            << ERROR_SD << "\t";
        fout << setw(8) << setprecision(4) << fixed << right
            << LVSIGMA << "\t";
        fout << setw(8) << setprecision(4) << fixed << right
            << WARCOSTMAX << "\t";
        fout << setw(8) << setprecision(4) << fixed << right
            << WCDISPAR << "\t";
        fout << setw(8) << setprecision(4) << fixed << right
            << REPAR << "\t";
        fout << setw(8) << setprecision(4) << fixed << right
            << META_SD << "\t";
    }
    if (style == 2)
    {
        fout << "***********************" << endl << endl;
        fout << "\t" << " PARAMETERS" << endl << endl;
        fout << "\t" << " " << endl << endl;
        fout << setw(8) << setprecision(4) << fixed << right
            << "POW_SD = " << "\t" << POW_SD << endl;
        fout << setw(8) << setprecision(4) << fixed << right
            << "ERROR_SD = " << "\t" << ERROR_SD << endl;
        fout << setw(8) << setprecision(4) << fixed << right
            << "LVSIGMA = " << "\t" << LVSIGMA << endl;
        fout << setw(8) << setprecision(4) << fixed << right
            << "WARCOSTMAX = " << "\t" << WARCOSTMAX << endl;
        fout << setw(8) << setprecision(4) << fixed << right
            << "WCDISPAR = " << "\t" << WCDISPAR << endl;
        fout << setw(8) << setprecision(4) << fixed << right
            << "REPAR = " << "\t" << REPAR << endl;
        fout << setw(8) << setprecision(4) << fixed << right
            << "META_SD = " << "\t" << META_SD << endl;
    }
}
#include "actper.h"
#include "actconc.h"
#include "actpol.h"
#include "actpolh.h"
#include "actutil.h"

void print_polarity(ostream &fout, int system_polarity, int major_size, int minor_size, int total_size)
{
    fout << setw(4) << right << system_polarity << "  ";
    fout << setw(2) << right << major_size << "  ";
    fout << setw(2) << right << minor_size << "  ";
    fout << setw(2) << right << total_size << "  ";
}

void print_eu(ostream &fout, int i_major, int t_major, float i_power, float t_power, float bigger, float p_ratio, float dcon, float i_eu, float t_eu, float low_eu_ratio, float high_eu_ratio, float str_int, int pinitiator_wot)
{
    fout << setw(1) << right << i_major << "  ";
    fout << setw(1) << right << t_major << "  ";
    fout << setw(6) << setprecision(3) << fixed << right << i_power << "  ";
    fout << setw(6) << setprecision(3) << fixed << right << t_power << "  ";
    fout << setw(6) << setprecision(3) << fixed << right << bigger << "  ";
    fout << setw(12) << setprecision(3) << fixed << right << p_ratio << "  ";
    fout << setw(12) << setprecision(3) << fixed << right << dcon << "  ";
    fout << setw(8) << setprecision(3) << fixed << right << i_eu << "  ";
    fout << setw(8) << setprecision(3) << fixed << right << t_eu << "  ";
    fout << setw(8) << setprecision(3) << fixed << right << low_eu_ratio << "  ";
    fout << setw(8) << setprecision(3) << fixed << right << high_eu_ratio << "  ";
    fout << setw(8) << setprecision(3) << fixed << right << str_int << "  ";
    fout << setw(1) << right << pinitiator_wot << "  ";
}

void print_run_round(ostream &fout, int run, int round)
{
    fout << setw(4) << fixed << right << run << "  ";
    fout << setw(4) << fixed << right << round << "  ";
}

void print_real(ostream &fout, int real_trade, int real_war)
{
    fout << setw(1) << right << real_trade << "  ";
    fout << setw(1) << right << real_war << "  ";
}

// Print out the number of run and round
// void print_run_round(ostream &fout, int run, int round)
{
    fout << setw(4) << fixed << right << run << "  ";
    fout << setw(4) << fixed << right << round << "  ";
}

// Print out the polarity value, major state number, minor state number
// void print_polarity(ostream &fout, int system_polarity, int major_size, int minor_size, int total_size)
{
    fout << setw(4) << right << system_polarity << "  ";
    fout << setw(2) << right << major_size << "  ";
    fout << setw(2) << right << minor_size << "  ";
    fout << setw(2) << right << total_size << "  ";
}

// Print out the expected utility of trade for initiator and target
// void print_eu(ostream &fout, int i_major, int t_major, float i_power, float t_power, float bigger, float p_ratio, float dcon, float i_eu, float t_eu, float low_eu_ratio, float high_eu_ratio, float str_int, int pinitiator_wot)
{
    fout << setw(1) << right << i_major << "  ";
    fout << setw(1) << right << t_major << "  ";
    fout << setw(6) << setprecision(3) << fixed << right << i_power << "  ";
    fout << setw(6) << setprecision(3) << fixed << right << t_power << "  ";
    fout << setw(6) << setprecision(3) << fixed << right << bigger << "  ";
    fout << setw(12) << setprecision(3) << fixed << right << p_ratio << "  ";
    fout << setw(12) << setprecision(3) << fixed << right << dcon << "  ";
    fout << setw(8) << setprecision(3) << fixed << right << i_eu << "  ";
    fout << setw(8) << setprecision(3) << fixed << right << t_eu << "  ";
    fout << setw(8) << setprecision(3) << fixed << right << low_eu_ratio << "  ";
    fout << setw(8) << setprecision(3) << fixed << right << high_eu_ratio << "  ";
    fout << setw(8) << setprecision(3) << fixed << right << str_int << "  ";
    fout << setw(1) << right << pinitiator_wot << "  ";
}

// Print out the variable values for real trade or war happening
// (0 = not really happened; 1 = really happened)
// void print_real(ostream &fout, int real_trade, int real_war)
{
    fout << setw(1) << right << real_trade << "  ";
    fout << setw(1) << right << real_war << "  ";
}
// Print out war size value briefly
void print_war_size_brief(ostream& fout, int war_size)
{
  fout << setw(2) << right << war_size << endl;
}

// Print out the information about initiator and target
void print_screen_l(ostream& cout, int run, int round, int number_of_states)
{
  cout << "S = " << setw(4) << right << run << " "
  << "R = " << setw(4) << right << round << " "
  << "N = " << setw(2) << right << number_of_states << " ";
}

// Print out the information about who wins
void print_screen_2(int war_or_trade, int index)
{
  if (war_or_trade == 1) cout << "TRADE :
else if (war_or_trade == 2) cout << " WAR :
else cout << " NONE :
  if (index == -1) cout << "No initiator selected" << endl;
  if (index == 0) cout << "War not implemented." << endl;
  if (index == 1) cout << "Initiator wins" << endl;
  if (index == 2) cout << "Target wins" << endl;
  if (index == 3) cout << "Trade happens" << endl;
}

// Print out the initial setup information to a log file
void print_initial(ostream& fout, int run, int round, int number_of_states,
                  vector<State> & vs, vector<Cell> & vc)
{
  fout << endl;
  fout << "*******************************************************" << endl;
  fout << run << " " << round << "\t" << "(State # = " << number_of_states << ", " << "\t" << "Cell # = " << vc.size() << ")" << endl << endl;
  for (int i = 0; i < vs.size(); i++)
    fout << vs.at(i).get_name() << ";
    fout << "POW:";
    fout << setw(10) << fixed << setprecision(4) << vs.at(i).get_state_power() << " ";
    fout << "URA:";
    fout << setw(10) << fixed << setprecision(4) << vs.at(i).get_state_ura() << " ";
    fout << "URB:";
    fout << setw(10) << fixed << setprecision(4) << vs.at(i).get_state_urb() << " ";
    fout << "(N = ";
    if (vs.at(i).get_sc_size() < 10)
```cpp
fout << setfill(' ') << setw(2) << vs.at(i).get_sc_size() << endl;
else
    fout << vs.at(i).get_sc_size() << endl;
for(int j = 0; j < vs.at(i).get_sc_size(); j++)
{
    int index = vs.at(i).get_sc(j);
    fout << "(";
    if (vc[index].get_position() < 10)
        fout << setfill(' ') << setw(2) << vc[index].get_position();
    else
        fout << vc[index].get_position();
    fout << "= " << vc[index].get_whether_capital() << ");"
    if ((j > 0) && ((j+1)%5 == 0) && ((vs.at(i).get_sc_size() - j) > 1))
        fout << endl << "\" << "\t" << "\t" << "\t" << "\t" << "\t"
            << "\t" << "\t" << "\t" << "\t" << "\t"
            << "\t" << "\t" << "\t" << "\t" << "\t"
            << "\t" << "\t" << "\t" << "\t" << "\t";
    else
        fout << " ";
}
fout << endl;
}
fout << endl;

// Print out the information about initiator and target
//====================================================================================================
void print_it(ostream& fout, int run, int round, int number_of_states, vector<State> vs, vector<Cell> vc,
              int pinitiator, int ptarget)
{
    // Printing out initiator information
    fout << "Initiator: " << vs[pinitiator].get_name() << " ";
    fout << "Power: ";
    fout << setw(10) << fixed << setprecision(4) << vs[pinitiator].get_state_power() << " ";
    fout << "URA: ";
    fout << setw(10) << fixed << setprecision(4) << vs[pinitiator].get_state_ura() << " ";
    fout << "URB: ";
    fout << setw(10) << fixed << setprecision(4) << vs[pinitiator].get_state_urb() << " ";
    fout << "(N = ");
    if (vs[pinitiator].get_sc_size() < 10)
        fout << setfill(' ') << setw(2) << vs[pinitiator].get_sc_size() << endl;
    else
        fout << vs[pinitiator].get_sc_size() << endl;
    for(int i = 0; i < vs[pinitiator].get_sc_size(); i++)
```
int index = vs[pinitiator].get_sc(i);
fout << "\n";
if (vc[index].get_position() < 10)
    fout << setfill(' ') << setw(2)
    << vc[index].get_position();
else
    fout << vc[index].get_position();
fout << "=" << vc[index].get_whether_capital() << "\n";
if ((i > 0) && ((i+1)%5 == 0)
    && (vs[pinitiator].get_sc_size() - i > 1))
    fout << endl << "|\t" << "|\t" << "|\t" << "|\t"
    << "|\t" << "|\t" << "|\t" << "|\t";
else
    fout << "\n";
}
fout << endl;

// Print out target information
// ------------------------------------------------------------------------
// Printing out target information
// ------------------------------------------------------------------------
fout << "Target: " << endl;
fout << vs[ptarget].get_name() << "\n";
fout << "POW:";
fout << setw(10) << fixed << setprecision(4)
    << vs[ptarget].get_state_power() << "\n";
fout << "URA:";
fout << setw(10) << fixed << setprecision(4)
    << vs[ptarget].get_state_ura() << "\n";
fout << "URB:";
fout << setw(10) << fixed << setprecision(4)
    << vs[ptarget].get_state_urb() << "\n";
if (vs[ptarget].get_sc_size() < 10)
    fout << setfill('.') << setw(2)
    << vs[ptarget].get_sc_size() << "|\t";
else
    fout << vs[ptarget].get_sc_size() << "|\t";
for (i = 0; i < vs[ptarget].get_sc_size(); i++)
{
    int index = vs[ptarget].get_sc(i);
fout << "\n";
if (vc[index].get_position() < 10)
    fout << setfill('.') << setw(2)
    << vc[index].get_position();
else
    fout << vc[index].get_position();
fout << "=" << vc[index].get_whether_capital() << "\n";
if ((i > 0) && ((i+1)%5 == 0) &&
    (vs[ptarget].get_sc_size() - i > 1))
    fout << endl << "|\t" << "|\t" << "|\t" << "|\t"
    << "|\t" << "|\t" << "|\t" << "|\t";
else
    fout << "\n";
}
fout << endl << endl;

// Print out no war or trade event information to a log file
// ------------------------------------------------------------------------
void print_no_event(ostream& fout, int run, int round, int number_of_states)
// Print out no war information to a log file
// ---------------------------------------------------------
void print_no_war(ostreams fout, int run, int round, int number_of_states)
{
    fout << "---------------------------------------------------------"
    << endl;
    fout << run << "-" << round << "\t"
    << "(# " << number_of_states << ")" << endl << endl;
    fout << "  ************************************ " << endl;
    fout << "  *  *  " << endl;
    fout << "  *  CASE: NO WAR NO TRADE *" << endl;
    fout << "  *  *  " << endl;
    fout << "  ************************************ " << endl;
}

// Print out trade information to a log file
// ---------------------------------------------------------
void print_trade(ostreams fout, int run, int round, int number_of_states)
{
    fout << "---------------------------------------------------------"
    << endl;
    fout << run << "-" << round << "\t"
    << "(# " << number_of_states << ")" << endl << endl;
    fout << "  **********************.************•• " << endl;
    fout << "  *  *  " << endl;
    fout << "  *  CASE: TRADE " << endl;
    fout << "  *  *  " << endl;
    fout << "  ************************************ " << endl;
}

// Print out trade information to a log file (details)
// ura_or_urb: 0 (A exports ura) 1 (A exports urb) 2 (No trade)
// ---------------------------------------------------------
void print_trade_details(ostreams fout, vector<State> & vs, vector<Cell> & vc, int a, int b, float unit, float price, int ura_or_urb)
{
    if (ura_or_urb == 2)
    fout << "No trade between state "
    << vs[a].get_name() << ", and state "
    << vs[b].get_name() << " due to small volume or non-complementary MRS's"
    << endl;
    else if (ura_or_urb == 0)
if (price >= 1.0)
{
    fout << "State " << vs[a].get_name() << " exchanged " << setw(6) << setprecision(3) << fixed << unit << " units of URA with " << price*unit << " units of URR from state " << vs[b].get_name() << endl;
} else
{
    fout << "State " << vs[a].get_name() << " exchanged " << setw(6) << setprecision(3) << fixed << unit / price << " units of URA with " << unit << " units of URB from state " << vs[b].get_name() << endl;
}
else
    // if ura_or_urb == 1
{
    if (price >= 1.0)
    {
        fout << "State " << vs[a].get_name() << " exchanged " << setw(6) << setprecision(3) << fixed << unit / price << " units of URA with " << unit << " units of URB from state " << vs[b].get_name() << endl;
    } else
    {
        fout << "State " << vs[a].get_name() << " exchanged " << setw(6) << setprecision(3) << fixed << unit / price << " units of URA with " << unit << " units of URB from state " << vs[b].get_name() << endl;
    }
}

// ==============================================================
// Print out alliance configuration (for each run)
// ==============================================================
void print_alliance(ostream& fout, int run, int round, int number_of_states,
vector<State> & vs, vector<int> offensive_part,
float opower, int pinitiator, vector<int> defensive_part,
float dpower, int ptarget)
{
    // Offense part printing
    fout << endl;
    fout << "Offensive part powers = " << opower << endl;
    fout << "Offensive part (" << vs[pinitiator].get_name() << ") = " << ";
    for (int i = 0; i < offensive_part.size(); i++)
    {
        fout << vs[offensive_part[i]].get_name() << " ";
        if ((i > 0) && ((i+1)%5 == 0)
            fout << endl << "\t" << "\t" << "\t";
        else
            fout << " ";
    }
fout << endl << endl;

// Defensive part printing
fout << "Defensive part powers = " << dpower << endl;
for (i = 0; i < defensive_part.size(); i++)
{
    fout << vs[defensive_part[i]].get_name() << " ";
    if ((i > 0) && ((i+1)%5 == 0) && (defensive_part.size() - i) > 1)
        fout << endl << "\t" << "\t" << "\t";
    else
        fout << " ";
}
fout << endl << endl;

// Print out lt, lv and rnd
void print_lvt(ostream& fout, int run, int round, int number_of_states,
               float lv, float lt, float rnd)
{
    fout << "---------------------------------------------------------" << endl;
    fout << run << "-" << round << "\t"
         << "(# " << number_of_states << ")" << endl << endl;
    fout << "LV = " << setw(8) << fixed << setprecision(5) << lv << "\t";
    fout << "LT = " << setw(8) << fixed << setprecision(5) << lt << "\t";
    fout << "RND = " << setw(8) << fixed << setprecision(5) << rnd << "\t";
    fout << endl << endl;
}

// Print out who wins
// Index = 1: initiator wins
// Index = 2: target wins
// Index = 3: tied
void print_who_wins(ostream& fout, int index)
{
    if (index == 1)
    {
        fout << "*****************************************************************************" << endl;
        fout << "* CASE: WAR -- INITIATOR WINS *" << endl;
        fout << "*****************************************************************************" << endl;
    }
    if (index == 2)
    {
        fout << "*****************************************************************************" << endl;
        fout << "* CASE: WAR -- TARGET WINS *" << endl;
        fout << "*****************************************************************************" << endl;
    }
}
if (index == 3)
{
    fout << "***************************************************************************" << endl;
    fout << " + CASE: WAR -- TIED +" << endl;
    fout << "***************************************************************************" << endl;
    fout << endl;
}
fout << endl;

// Print out cell claim vector
//
// void print_cell_claim(ostream& fout, int cell_loss, vector<State> & vs,
// vector<int> victor_list, vector<float> victor_power,
// vector<float> victor_power_ratio, vector<int> precell_claim,
// vector<int> intcell_claim)
{
    fout << "CELL CLAIM CALCULATION" << endl;
    fout << "***************************************************************************" << endl;
    fout << "STATE " "POWER " "RATIO " "PRECELL " "INTCELL" << endl;
    fout << "------------------" "----------" "--------" "-------" << endl;
    int diff = precell_claim.size() - intcell_claim.size();
    for (int a = 0; a < precell_claim.size(); a++)
    {
        fout << " " << vs[victor_list[a]].get_name() << " ";
        fout << setw(10) << fixed << setprecision(5) << victor_power[a] << " ";
        fout << setw(8) << fixed << setprecision(5) << victor_power_ratio[a] << " ";
        fout << setw(4) << setprecision(0) << precell_claim[a] << " ";
        if (a - diff >= 0)
        {
            fout << setw(4) << setprecision(0) << intcell_claim[a-diff] << endl;
        }
        else
        {
            fout << " -" << endl;
        }
    }
    fout << "------------------" "----------" "--------" "-------" << endl;
    fout << "Total cell loss = " << cell_loss << endl;
    fout << "***************************************************************************" << endl;
}

// Print out cell vector elements
//
// void print_cell_vector(ostream& fout, vector<Cell> & vc,
// vector<int> cell_vector, string s)
{
    fout << s ;
    if (s.size() < 8) fout << "\t\t\t\t";
    if ((s.size() >= 8) & (s.size() < 16)) fout << "\t\t\t";
    322
if (s.size() >= 16) && (s.size() < 24) fout << "\t\t";
if (s.size() >= 24) && (s.size() < 32) fout << "\t";
if (s.size() >= 32) fout << "\";
for (int i = 0; i < cell_vector.size(); i++)
{
    fout << "(" << vc[cell_vector[i]].get_position() << ")";
    if ((i > 0) && ((i+1)%5 == 0) && (cell_vector.size() - i > 1))
        fout << endl << "\t\t\t\t";
    else
        fout << "\";
}
fout << endl;

// Print out cell transfer
void print_transfer(ostream& fout, vector<State> & vs, vector<Cell> & vc, int pcell, int ploser, int claimant)
{
    fout << "State " << vs[ploser].get_name() << " transferred the following cells to claim state " << vs[claimant].get_name() << ": " "(" << vc[pcell].get_position() << ")" "endl;
}

// Print out the generation of a new state after cell transfer
void print_new_state(ostream& fout, vector<State> & vs, vector<Cell> & vc, int ploser, int pnew, int pcell)
{
    fout << "State " << vs[ploser].get_name() << " transferred the following cells to a new state " << vs[pnew].get_name() << ": " "(" << vc[pcell].get_position() << ")" "endl;
}

// Print out the size of war
void print_war_size(ostream& fout, int run, int round, int osize, int dsize, float ipower, float tpower, float opower, float dpower, float totalpower, int warcells)
{
    fout << setw(4) << right << run << "\t"
    << setw(4) << right << round << "\t"
    << setw(4) << right << osize << "\t"
    << setw(4) << right << dsize << "\t"
    << setw(4) << right << warcells << "\t"
    << setw(9) << right << setprecision(6) << ipower << "\t"
    << setw(9) << right << setprecision(6) << tpower << "\t"
    << setw(9) << right << setprecision(6) << opower << "\t"
    << setw(9) << right << setprecision(6) << dpower << "\t"
    << setw(6) << right << fixed << setprecision(1) << totalpower
    << endl;
}
void print_result(ostream& fout, int new_state_number, int trade_freq,
    int no_war_freq, int war_freq, int target_wins,
    int initiator_wins, int real_round, int final_state_number)
{
    fout << setw(4) << right << new_state_number << "\t";
    fout << setw(4) << right << trade_freq << "\t";
    fout << setw(4) << right << no_war_freq << "\t";
    fout << setw(4) << right << war_freq << "\t";
    fout << setw(4) << right << target_wins << "\t";
    fout << setw(4) << right << initiator_wins << "\t";
    fout << setw(4) << right << real_round << "\t";
    fout << setw(4) << right << final_state_number << "\t";
    fout << endl;
}

void print_world_total_power(ostream& fout, int run, int round,
    int number_of_states, float world_total_power)
{
    fout << setw(4) << right << run << "\t"
        << setw(4) << right << round << "\t"
        << setw(2) << right << number_of_states << "\t"
        << setw(13) << fixed << right << setprecision(5)
        << world_total_power << endl;
}
// State.h
// Declaration of the State class
// Written Feb. 14, 2000
// Revised Mar. 07, 2000
// Revised Apr. 12, 2000
// Revised May 30, 2000
//

#ifndef _STATE_H_
#define _STATE_H_

#include <vector>
#include <list>
#include <algorithm>
#include <iostream>
#include <string>
#include "Param.h"
#include "Cell.h"

using namespace std;

class Cell; // Forward declaration

//========================================================================
// Structure definitions
//========================================================================

// Combined data structure of a state's max. util and its target
struct state_eu
{
  int id;
  float real_power;
  float estimated_power;
  float max_util;
  int target;
  int wt; // 0 = no action, 1 = trade, 2 = war
  float target_power;
  float lv;
  float relative_real_power;
  float weighted_eu; // considering power difference
};

// Combined data structure of a state's util toward one neighbor
struct neighbor_eu
{
  int id; // target id
  float power; // target's power
  float eut; // utility of trade
  float euw; // utility of war
  float bigger; // bigger utility
  int target; // target of bigger utility
  float lv; // likelihood of war victory
  int wt; // 0 = no action, 1 = trade, 2 = war
};

struct major_power
{
```cpp
int id;
float power;

// Overloaded operator definition for major power size sorting
bool operator<(major_power a, major_power b);

// Class declaration of State
class State
{
private:
  int id; // s[id] in a current world
  string name; // 0 = not major; 1 = major
  vector<int> sc;
  vector<int> snc;
  vector<int> sns;
  float state_power;
  float state_ura;
  float state_urb;
  float per_mà;
  float per_mb;
  int capital; // c[capital] as the capital index

public:
  // Constructors and destructors
  State(); // Default constructor
  State(int n, int a, vector<Cell> & vc, vector<State> & vs);
  State(int n, vector<int> cv, vector<Cell> & vc, vector<State> & vs);
  State(const State& rs); // Copy constructor
  State& operator=(const State& rs); // Assignment operator
  ~State(); // Destructor

  // State id and major status setting
  void set_id(int n) { id = n; }
  int get_id() { return id; } // Returns state is - s[id]
  string get_name() { return name; } // Shows state name
  void set_major(int n) { major = n; }
  int get_major() { return major; }
  void set_name(int n); // Sets state id number

  // State cell functions
  int get_sc(int a) { return sc[a]; }
  int get_sc_size() { return sc.size(); }
  int get_sc_position(int a) { return sc[a] + 1; } // sc[a]'s position
  void sc_add(vector<State> & vs, vector<Cell> & vc, int a);
  void sc_temp_add(vector<State> & vs, vector<Cell> & vc, int a);
  void sc_delete(vector<State> & vs, vector<Cell> & vc, int a);
  void sc_temp_delete(vector<State> & vs, vector<Cell> & vc, int a);
};
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
bool has_capital(); // 1 if state has capital, 0 otherwise
int get_capital(); // Returns the index of the capital cell
void set_capital(vector<Cell> & vc, int a); // Set a cell as capital
bool cells_contiguous(vector<State> & vs, vector<Cell> & vc); // Whether all cells of a state are contiguous or not
int get_block_number(vector<Cell> & vc);

// State neighbor cells functions
int get_snc(int a)
int get_snc_size()
int get_snc_position(int a)
void find_snc(vector<Cell> & vc, int a); // Find if vc[a] is in snc

// State neighbor states functions
int get_sns(int a)
int get_sns_size()
void set_sns(vector<Cell> & vc);

// State power functions
void set_state_power(float p)
float get_state_power()
void reduce_state_power(float rate, vector<Cell> & vc);
void subtract_state_power(float volume, vector<Cell> & vc);
void add_state_power(float volume, vector<Cell> & vc);
void cal_state_power(vector<Cell> & vc); // Calculate and assign it

// State resource and metabolism functions
void set_state_ura(float a)
void set_state_urb(float b)
float get_state_ura()
float get_state_urb()
float get_per_ma()
float get_per_mb()
void set_per_ma(float a);
void set_per_mb(float b);
void cal_state_ura(vector<Cell> & vc);
void cal_state_urb(vector<Cell> & vc);
void add_state_ura(vector<Cell> & vc, float a);
void add_state_urb(vector<Cell> & vc, float b);
void reduce_state_ura(vector<Cell> & vc, float a);
void reduce_state_urb(vector<Cell> & vc, float b);

#endif // _STATE_H_
#include "State.h"
#include <cmath>
using namespace std;

class Cell; // Forward class definition

// Function predicate for sorting major_power vector
// From the major_power holding biggest power to lowest
bool operator<(major_power a, major_power b)
{
    return a.power > b.power;
}

// Constructors and destructors
State::State() :
    id(-1),
    name(""),
    major(0),
    state_power(0.0),
    state_ura(MINIMUM_UR),
    state_urb(MINIMUM_UR),
    per_ma(MINIMUM_ETA),
    per_mb(MINIMUM_ETA),
    capital(-1)
{
    vector<int> sc(0); // vector of state cell references
    vector<int> snc(0); // vector of state neighbor cell references
    vector<int> sns(0); // vector of state neighbor state references
}

State::State(int n, int a, vector<Cell> & vc, vector<State> & vs)
    :
    id(vs.size()),
    name(""),
    major(0),
    state_power(0.0),
    state_ura(MINIMUM_UR),
    state_urb(MINIMUM_UR),
    per_ma(MINIMUM_ETA),
    per_mb(MINIMUM_ETA),
    capital(a)
{
    set_name(n);
    vector<int> sc(0); // vector of state cell references
    vector<int> snc(0); // vector of state neighbor cell references
    vector<int> sns(0); // vector of state neighbor state references
    sc_add(vs, vc, a);
    // Rest of the code...

set_snc(vc);
set_sns(vc);

State::State(int n, vector<int> cv, vector<Cell> & vc, vector<State> & vs)
: id(vs.size()),
  name(""),
  major(0),
  state_power(0.0),
  state_ura(MINIMUM_UR),
  state_urb(MINIMUM_UR),
  per_ma(MINIMUM_META),
  per_mb(MINIMUM_META),
  capital(-1) // Will be determined below
{
  set_name(n);
  vector<int> sc(0); // vector of state cell references
  vector<int> snc(0); // vector of state neighbor cell references
  vector<int> sns(0); // vector of state neighbor state references
  for (int i = 0; i < cv.size(); i++)  sc_add(vs, vc, cv.at(i));
  this->cal_state_power(vc);
  this->cal_state_ura(vc);
  this->cal_state_urb(vc);
  this->set_snc(vc);
  this->set_sns(vc);
}

State::State(const States rs) : id(rs.id),
  name(rs.name),
  major(rs.major),
  state_power(rs.state_power),
  state_ura(rs.state_ura),
  state_urb(rs.state_urb),
  per_ma(rs.per_ma),
  per_mb(rs.per_mb),
  capital(rs.capital),
  sc(rs.sc),
  snc(rs.snc),
  sns(rs.sns)
{
}

State& State::operator=(const State& rs)
{
  id = rs.id;
  name = rs.name;
  major = rs.major;
  state_power = rs.state_power;
  state_ura = rs.state_ura;
  state_urb = rs.state_urb;
  per_ma = rs.per_ma;
  per_mb = rs.per_mb;
  capital = rs.capital;
  sc = rs.sc;
  snc = rs.snc;
  sns = rs.sns;
  return *this;
}

State::~State()
{
sc.clear();
snc.clear();
sns.clear();

// State id and major status setting
// ---------------------------------------------
void State::set_name(int n)    // n (0 - 97)
{
    int il;
    int i2;
    il = n / 26;
    i2 = (n % 26);
    char starter = 'A';
    char bufferl = starter + il;
    char buffer2 = starter + i2;
    if (!name.empty())
    {
        int size_name = name.size();
        name.erase(0, size_name);  // Make string empty if there is any
    }
    name.insert(0, buffer2);
    name.insert(0, bufferl);
}

// State cell functions
// ---------------------------------------------
void State::sc_add(vector<State> & vs, vector<Cell> & vc, int a)
{
    if (this->get_sc_size() == 0)
    {
        vc[a].set_whether_capital(1);
    }

    if (this->has_capital())
    {
        vc[a].set_whether_capital(0);
    }
    else
    {
        vc[a].set_whether_capital(1);
        this->set_capital(vc, a);
    }
    vc[a].set_which_state(this->get_id());

    sc.push_back(a);
    this->cal_state_ura(vc);
    this->cal_state_urb(vc);
    this->cal_state_power(vc);
    this->set_snc(vc);
    this->set_sns(vc);
}

void State::sc_temp_add(vector<State> & vs, vector<Cell> & vc, int a)
sc.push_back(a);
vc[a].set_which_state(this->get_id());
this->set_snc(vc);
this->set_sns(vc);
}

void State::sc_delete(vector<State> & vs, vector<Cell> & vc, int a)
{
    vector<int>::iterator iter;
    iter = find(sc.begin(), sc.end(), a);
    if (iter != sc.end())
    {
        sc.erase(iter);
        if (vc[a].get_which_state() == 1)
        {
            if (this->get_sc_size() > 0) // set a new capital for loser
            {
                vc[this->get_sc(0)].set_which_state(1);
                this->set_capital(vc, this->get_sc(0));
            }
        }
        vc[a].set_which_state(-1); // Null value
        if (this->sc.size() > 0)
        {
            this->cal_state_ura(vc);
            this->cal_state_urb(vc);
            this->cal_state_power(vc);
        }
        else // Should not reach here
        {
            cerr << " (in State::sc_delete()) No cell in sc." << endl;
            exit(1);
        }
    }
}

void State::sc_temp_delete(vector<State> & vs, vector<Cell> & vc, int a)
{
    vector<int>::iterator iter;
    iter = find(sc.begin(), sc.end(), a);
    if (iter != sc.end())
    {
        sc.erase(iter);
        vc[a].set_which_state(-1);
        this->set_snc(vc);
        this->set_sns(vc);
    }
    else // Should not reach here
    {
        cerr << " (in State::sc_temp_delete()) No cell in sc." << endl;
        exit(1);
    }
}
bool State::has_capital()
{
    if (capital >= 0) return true;
    else return false;
}

int State::get_capital()
{
    if (capital >= 0)
        return capital;
    else
    {
        cerr << "This state does not have any capital cell."
             << " in State::get_capital() function." << endl;
        exit(1);
    }
}

void State::set_capital(vector<Cell> & vc, int a)
{
    capital = vc[a].get_id();
}

bool State::cells_contiguous(vector<State> & vs, vector<Cell> & vc)
{
    int total_cells = this->get_sc_size();
    if (total_cells > 1)
    {
        for (int i = 0; i < this->get_sc_size() - 1; i++)
        {
            int indexi = this->get_sc(i);
            vector<vector<int>> contiguous_vv;
            contiguous_vv = vc[indexi].set_contiguous_vv(vs, vc);
            if (contiguous_vv.size() != total_cells)
                return false;
        }
        return true;
    }
    else // If total_cells = 0 or 1
        return true;
}

int State::get_block_number(vector<Cell> & vc)
{
    if (this->get_sc_size() == 0)
    {
        cerr << "This state does not have any cell." << endl;
        exit(1);
    }
    if (this->get_sc_size() == 1) return 1;
    vector<int> remains;
    vector<vector<int>> conti_blocks; // its size will be returned
    for (int i = 0; i < this->get_sc_size(); i++)
        remains.push_back(this->get_sc(i));
    while (remains.size() > 0)
    {
        int pc = remains.back();
        remains.pop_back();
        if (conti_blocks.size() == 0)
vector<int> temp;
temp.push_back(pc);
conti_blocks.push_back(temp);
continue;  // Go back to while loop
}
else
{
    int sentinel = 0;
    for (int a = 0; a < conti_blocks.size(); a++)
    {
        vector<int> temp = conti_blocks[a];
        for (int b = 0; b < temp.size(); b++)
        {
            int pcref = temp[b];
            if (vc[pcref].is_neighbor_cell(vc, pc))
            {
                temp.push_back(pc);
                sentinel = 1;
                break;
            }
        }
        if (sentinel == 1) continue;
    }
    // If the cell is not contiguous to any cells
    vector<int> another;
    another.push_back(pc);
    conti_blocks.push_back(another);
}
// End of while
return conti_blocks.size();

// State neighbor cells functions
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = :
void State::set_snc(vector<Cell> & vc)
{
    snc.clear();
    for(int i = 0; i < get_sc_size(); i++)
    {
        int indexi = this->get_sc(i);
        for(int j = 0; j < 6; j++)
        {
            int indexj = vc[indexi].get_neighbor_cell(j);
            if (indexj == -1) continue;
            vector<int>::iterator iter1, iter2;
            iter1 = find(sc.begin(), sc.end(), indexj);
            iter2 = find(snc.begin(), snc.end(), indexj);
            // No duplicate cell reference and no sc member
            if ((iter1 == sc.end()) && (iter2 == snc.end()))
                snc.push_back(indexj);
        }
    }
}

bool State::find_snc(vector<Cell> & vc, int a)
{
    this->set_snc(vc);
    this->set_sns(vc);
    for(int i = 0; i < this->get_snc_size(); i++)

if (snc[i] == vc[a].get_id())
    return true;
return false;

// State neighbor states functions
void State::set_sns(vector<Cell> & vc)
{
    sns.clear();
    for(int i = 0; i < this->get_snc_size(); i++)
    {
        if (this->get_snc(i) == -1)
            continue;
        else
        {
            int cell_index = this->get_snc(i);
            int state_index = vc[cell_index].get_which_state();
            vector<int>::iterator iter;
            iter = find(sns.begin(), sns.end(), state_index);
            if(iter == sns.end() || (state_index != this->get_id()))
                sns.push_back(state_index);
        }
    }
}

// State power functions
void State::reduce_state_power(float rate, vector<Cell> & vc)
{
    // Calculate the ratio between ura and urb
    float volume = this->get_state_power()*rate;
    float ura = this->get_state_ura();
    float urb = this->get_state_urb();
    float a_less = (ura/(ura+urb))*volume;
    float b_less = (urb/(ura+urb))*volume;
    // Distribute the burdens to each cell
    for (int i = 0; i < this->get_sc_size(); i++)
    {
        int acell = this->get_sc(i);
        float a = (vc[acell].get_ura()/ura)*a_less;
        float b = (vc[acell].get_urb()/urb)*b_less;
        vc[acell].reduce_ura(a);
        vc[acell].reduce_urb(b);
    }
    // Recalculate the state_ura, state_urb, state_power
    this->cal_state_power(vc);
}

void State::subtract_state_power(float volume, vector<Cell> & vc)
{
    // Calculate the ratio between ura and urb
    float ura = this->get_state_ura();
    float urb = this->get_state_urb();
    float a_ratio = ura/(ura+urb);
    float b_ratio = urb/(ura+urb);
    float a_less = volume*a_ratio;
    float b_less = volume*b_ratio;
    // Distribute the burdens to each cell
for (int i = 0; i < this->get_sc_size(); i++)
{
    int acell = this->get_sc(i);
    float a = (vc[acell].get_ura()/ura)*a_less;
    float b = (vc[acell].get_urb()/urb)*b_less;
    vc[acell].reduce_ura(a);
    vc[acell].reduce_urb(b);
}
// Recalculate the state_ura, state_urb, state_power
this->cal_state_power(vc);

void State::add_state_power(float volume, vector<Cell> & vc)
{
    // Calculate the ratio between ura and urb
    float ura = this->get_state_ura();
    float urb = this->get_state_urb();
    float a_ratio = ura/(ura+urb);
    float b_ratio = urb/(ura+urb);
    float a_more = volume*a_ratio;
    float b_more = volume*b_ratio;
    // Distribute the burdens to each cell
    for (int i = 0; i < this->get_sc_size(); i++)
    {
        int acell = this->get_sc(i);
        float a = (vc[acell].get_ura()/ura)*a_more;
        float b = (vc[acell].get_urb()/urb)*b_more;
        vc[acell].add_ura(a);
        vc[acell].add_urb(b);
    }
    // Recalculate the state_ura, state_urb, state_power
    this->cal_state_power(vc);
}

void State::cal_state_power(vector<Cell> & vc)
{
    this->cal_state_ura(vc);
    this->cal_state_urb(vc);
    float ura = this->get_state_ura();
    float urb = this->get_state_urb();
    int n = this->get_sc_size();
    float ma = per_ma * n;
    float mb = per_mb * n;
    float a_element = pow(ura, ma/(ma+mb));
    float b_element = pow(urb, mb/(ma+mb));
    this->set_state_power(a_element*b_element);
}

// State resource and metabolism functions

// Set per Ma
void State::set_per_ma(float a)
{
    if ( a < MINIMUM_META)
        per_ma = MINIMUM_META;
    else
        per_ma = a;
}

void State::set_per_mb(float b)
{
if (b < MINIMUM_META)
    per_mb = MINIMUM_META;
else
    per_mb = b;
}

void State::cal_state_ura(vector<Cell> & vc)
{
    float temp = 0.0;
    if (this->get_sc_size() > 0)
    {
        for (int i = 0; i < this->get_sc_size(); i++)
        {
            int acell = this->get_sc(i);
            temp += vc[acell].get_ura();
        }
    }
    this->state_ura = temp;
}

void State::cal_state_urb(vector<Cell> & vc)
{
    float temp = 0.0;
    if (this->get_sc_size() > 0)
    {
        for (int i = 0; i < this->get_sc_size(); i++)
        {
            int acell = this->get_sc(i);
            temp += vc[acell].get_urb();
        }
    }
    this->state_urb = temp;
}

void State::add_state_ura(vector<Cell> & vc, float a)
{
    float total_ura = 0.0;
    for (int i = 0; i < this->get_sc_size(); i++)
    {
        int acell = this->get_sc(i);
        total_ura += vc[acell].get_ura();
    }
    for (i = 0; i < this->get_sc_size(); i++)
    {
        int acell = this->get_sc(i);
        float added = a * (vc[acell].get_ura()/total_ura);
        vc[acell].add_ura(added);
    }
}

void State::add_state_urb(vector<Cell> & vc, float b)
{
    float total_urb = 0.0;
    for (int i = 0; i < this->get_sc_size(); i++)
    {
        int acell = this->get_sc(i);
        total_urb += vc[acell].get_urb();
    }
    for (i = 0; i < this->get_sc_size(); i++)
    {
        int acell = this->get_sc(i);  

float added = b * (vc[acell].get_urb()/total_urb);
vc[acell].add_urb(added);
}

void State::reduce_state_ura(vector<Cell> & vc, float a) {
  float total_ura = 0.0;
  for (int i = 0; i < this->get_sc_size(); i++)
  {
    int acell = this->get_sc(i);
    total_ura += vc[acell].get_ura();
  }
  for (i = 0; i < this->get_sc_size(); i++)
  {
    int acell = this->get_sc(i);
    float reduced = a * (vc[acell].get_ura()/total_ura);
    vc[acell].reduce_ura(reduced);
  }
}

void State::reduce_state_urb(vector<Cell> & vc, float b) {
  float total_urb = 0.0;
  for (int i = 0; i < this->get_sc_size(); i++)
  {
    int acell = this->get_sc(i);
    total_urb += vc[acell].get_urb();
  }
  for (i = 0; i < this->get_sc_size(); i++)
  {
    int acell = this->get_sc(i);
    float reduced = b * (vc[acell].get_urb()/total_urb);
    vc[acell].reduce_urb(reduced);
  }
}
Transfer.h
Declaration of cell-transfer-related functions

#ifndef _TRANSFER_H_
#define _TRANSFER_H_

#include <vector>
#include <algorithm>
#include <iostream>
#include "Cell.h"
#include "State.h"

using namespace std;

// Calculate the integer vector of cells to be transferred from a float vector
vector<int> cal_precell_claim(int cell_loss, vector<float> victor_power_ratio);

// Fill out the cell_claim vector (non-zero elements)
vector<int> cal_cell_claim(vector<int> precell_claim);

// Check whether the cell_claim vector accumulates to the cell_loss
void check_cell_claim(ostream & fout, vector<float> victor_power_ratio,
                      vector<int> precell_claim, vector<int> cell_claim, int cell_loss);

// Fill out the cell_claimants states vector (claiming 1+ cells)
vector<int> cal_cell_claimants(vector<int> intcell_claim,
                               vector<int> victor_list, vector<State> & vs);

// Cell transfer function 1 (in the cell transfer module)
void cell_transfer1(vector<State> & vs, vector<Cell> & vc, int ploser,
                    int winner, int pcell, int & how_many);

// Cell transfer function 2 (in the rearrangement module)
void cell_transfer2(vector<State> & vs, vector<Cell> & vc,
                    int ploser, int winner, int pcell);

// Cell transfer function 3 (in the cell transfer module)
void cell_transfer3(vector<State> & vs, vector<Cell> & vc, int ploser,
                    int winner, int pcell);

// Temporary cell transfer function
void cell_temp_transfer(vector<State> & vs, vector<Cell> & vc, int pa, int pb, int pcell);

void make_nonsplit(vector<State> & vs, vector<Cell> & vc, int pwinner, int ploser, vector<int> & contiguous_cells, bool & cut);

int loser_border_length(vector<Cell> & vc, int ploser, int cell);

int winner_border_length(vector<Cell> & vc, int pwinner, int cell);

void make_loser_compact(int ploser, vector<Cell> & vc, vector<int> & contiguous_cells);

void make_winner_compact(int pwinner, vector<Cell> & vc, vector<int> & contiguous_cells);

void transfer_first_cell(vector<int> & cell_list, vector<State> & vs, vector<Cell> & vc, int ploser, int claimant, int how_many);

void new_state_transfer(int new_state_id, vector<int> cell_vector, vector<State> & vs, vector<Cell> & vc, int ploser, vector<int> & pool_new_states, int how_many, vector<int> & cell_list, ostream & fout, float META_SD);

void transfer_first_cell(vector<int> & cell_vector, vector<Cell> & vc, int ploser, int claimant, int how_many);

void new_state_transfer(int new_state_id, vector<int> cell_vector, vector<State> & vs, vector<Cell> & vc, int ploser, vector<int> & pool_new_states, int how_many, vector<int> & cell_list, ostream & fout, float META_SD);

Generate a new state and transfer the first cell of cell_vector from the loser to the new state

Generate a new state and transfer the parameter cell from the loser to the new state
void new_state_transfer_cell (int new_state_id, int pcell, vector<State> & vs, vector<Cell> & vc, int ploser, vector<int> & pool_new_states, int & how_many, vector<int> & cell_list, ostream & fout, float META_SD);

// Generate a new state and transfer the first cell of cell_vector
// from the loser to the new state
void new_state_transfer2(int new_state_id, int pcell, vector<State> & vs, vector<Cell> & vc, int ploser, float META_SD);

vector<vector<int> > make_contiguous_blocks(vector<Cell> & vc, vector<int> fragmented);

vector<int> choose_main_block(vector<vector<int> > v_conti_blocks);

bool is_cell_vector_contiguous (vector<State> & vs, vector<Cell> & vc, vector<int> cell_vector);

int get_cell_vector_block_number(vector<State> & vs, vector<Cell> & vc, vector<int> cell_vector);

#if 0
#endif // TRANSFER_H

340

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
#include <vector>
#include <cmath>
#include <iomanip>
#include <algorithm>
#include "Param.h"
#include "Transfer.h"
#include "Prints.h"
#include "Mathfunc.h"

using namespace std;

// Calculate the integer vector of cells to be transferred from a float vector
vector<int> cal_precell_claim(int cell_loss, vector<float> victor_power_ratio) {
    int number = victor_power_ratio.size();
    vector<int> precell_claim(number, 0);
    if ((cell_loss == 0) || (number == 0)) return precell_claim;
    vector<float>::iterator iter_max;
    iter_max = max_element(victor_power_ratio.begin(),
                             victor_power_ratio.end());

    // Rounding rule
    for (int i = 0; i < number; i++)
    {
        float temp1 = victor_power_ratio.at(i) * cell_loss;
        float temp2;
        double temp3; // for applying to modf() function
        // temp3 holds integer part, temp2 holds floating part
        temp2 = (float) modf(temp1, &temp3);
        int rec = (int) temp3; // Casting to integer
        if (temp2 >= 0.5)
            rec++; // Round up to nearest integer
        precell_claim.at(i) = rec;
    }

    // If the distribution of float values make one cell remain after its
    // conversion to integers, add that one to the most powerful state.
    // (e.g., cell_loss = 2, distribution = 1.1, 0.3, 0.3, 0.3; in this
    // case, the most powerful state will take two cells)
    int check_sum = 0;
    for (i = 0; i < precell_claim.size(); i++)
    {
        check_sum += precell_claim.at(i); // If cell_loss is zero, set every element zero
        if ((cell_loss == 0) && (cell_loss < check_sum))
            for (i = 0; i < precell_claim.size(); i++)
                precell_claim.at(i) = 0;
        // If cell_loss is bigger than zero and bigger than the check_sum
        // then assign one more from the back (the biggest state)
        if ((cell_loss > 0) && (cell_loss > check_sum))
            
        
    }
}

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
int index1 = number - 1;
while (check_sum != cell_loss)
{
    precell_claim.at(index1)++;
    check_sum++;
    index1--;
}

// If cell_loss is bigger than zero and less than the check_sum
// then reduce one from the front (the weakest state)
if ((cell_loss > 0) && (cell_loss < check_sum))
{
    int index2 = 0;
    while (check_sum != cell_loss)
    {
        if (precell_claim.at(index2) > 0)
        {
            precell_claim.at(index2)--;
            check_sum--;
        }
        index2++;
    }
}

// The sum of cell_recv may be less than cell_loss because of
// rounding error; (ex) Four states may have power ratios of
// 50%, 30%, 10%, 10% for 4 cell_loss; in this case, cell_recv
// should be 2, 1, 0, 0 for each state, so one is remaining.
// If the sum exceeds cell_loss, then the loop will stop at some
// point of distributing cell_loss, but this is natural as the
// weakest is excluded from the process.
return precell_claim;

vector<int> cal_cell_claim(vector<int> precell_claim)
{
    vector<int> return_vector;
    if (precell_claim.size() == 0)
        return return_vector;
    for (int i = 0; i < precell_claim.size(); i++)
        if (precell_claim.at(i) > 0)
            return_vector.push_back(precell_claim.at(i));
    return return_vector;
}

void check_cell_claim(ostream & fout, vector<float> victor_power_ratio,
        vector<int> precell_claim, vector<int> cell_claim, int cell_loss)
{
    // Exception handling
    int sum = 0;
    for (int i = 0; i < cell_claim.size(); i++)
        sum += cell_claim.at(i);
if (sum != cell_loss)
{
    fout << "**********************************************************************************
     " << endl;
    fout << "ERROR IN CELL_CLAIM CALCULATION" << endl;
    fout << "-----------------------------" << endl << endl;
    fout << "power_ratio " << "precell_claim " << "cell_claim"
     << endl;
    fout << "____________ " << "  " << " __________
     " << endl;
    int diff = precell_claim.size() - cell_claim.size();
    for (int a = 0; a < precell_claim.size(); a++)
    {
        fout << setw(ll) << fixed << setprecision(5);
        fout << victor_power_ratio.at(a) << " ";
        fout << setw(13) << setprecision(0);
        fout << precell_claim.at(a) << " ";
        if (a - diff >= 0)
        {
            fout << setw(10) << setprecision(0);
            fout << cell_claim.at(a-diff) << endl;
        }
        else fout << "  ";
        fout << endl;
    }
    fout << endl;
    fout << " cell_claim sum: " << sum << "  ";
    fout << " cell_loss: " << cell_loss << endl << endl;
    fout << "**********************************************************************************
     " << endl;
    cerr << "The cell claim does not sum up to cell_loss." << endl;
    exit(1);
}

private:

vector<int> cal_cell_claimants(vector<int> intcell_claim,
     vector<int> victor_list, vector<State> & vs)
{
    vector<int> return_vector;
    if (intcell_claim.size() == 0) return return_vector;
    else
    {
        int index = 0;
        while (index != intcell_claim.size())
        {
            return_vector.push_back(victor_list.back());
            victor_list.pop_back();
            index++;
        }
    }
    reverse(return_vector.begin(), return_vector.end());
    return return_vector;
}

void cell_transfer1(vector<State> & vs, vector<Cell> & vc, int ploser,
int pwinner, int pcell, int & how_many) {
    if (vc[pcell].get_which_state() != ploser) {
        cerr << "The cell to transfer does not belong to the loser "
             << endl;
        cerr << "in the cell_transfer() function."
             << endl;
        exit(1);
    }
    vs[ploser].sc_delete(vs, vc, pcell);
    vs[pwinner].sc_add(vs, vc, pcell);
    how_many--;

    // -----------------------------------------------
    // Cell-state connection resetting
    // -----------------------------------------------
    for (int j = 0; j < vs.size(); j++)
        for (int k = 0; k < vs[j].get_sc_size(); k++)
            (int pcell = vs[j].get_sc(k);
             vc[pcell].set_which_state(vs[j].get_id()));
    for (int m = 0; m < vs.size(); m++)
        (vs[m].set_snc(vc);
         vs[m].set_sns(vc);
    }
}

// -----------------------------------------------
// Cell transfer function 2 (in the rearrangement module)
// -----------------------------------------------
void cell_transfer2(vector<State> & vs, vector<Cell> & vc,
                   int ploser, int pwinner, int pcell) {
    if (vc[pcell].get_which_state() != ploser) {
        cerr << "The cell to transfer does not belong to the loser "
             << endl;
        cerr << "in the cell_transfer() function."
             << endl;
        exit(1);
    }
    vs[ploser].sc_delete(vs, vc, pcell);
    vs[pwinner].sc_add(vs, vc, pcell);
}

// -----------------------------------------------
// Cell transfer function 3 (in the cell transfer module)
// -----------------------------------------------
void cell_transfer3(vector<State> & vs, vector<Cell> & vc, int ploser,
                    int pwinner, int pcell) {
    if (vc[pcell].get_which_state() != ploser) {
        cerr << "The cell to transfer does not belong to the loser "
             << endl;
        cerr << "in the cell_transfer() function."
             << endl;
        exit(1);
    }
    vs[ploser].sc_delete(vs, vc, pcell);
    vs[pwinner].sc_add(vs, vc, pcell);
//print_transfer(fout, vs, vc, pcell, ploser, pwinner);

// Cell-state connection resetting
for (int j = 0; j < vs.size(); j++)
for (int k = 0; k < vs[j].get_sc_size(); k++)
{
    int pcell = vs[j].get_sc(k);
    vc[pcell].set_which_state(vs[j].get_id());
}
for (int m = 0; m < vs.size(); m++)
{
    vs[m].set_snc(vc);
    vs[m].set_sns(vc);
}

void cell_temp_transfer(vector<State> & vs, vector<Cell> & vc, int pa, int pb, int pcell)
{
    if (vc[pcell].get_which_state() != pa)
    {
        cerr << "The cell to transfer does not belong to the loser "
            << endl;
        cerr << "in the cell_temp_transfer() function." << endl;
        exit(1);
    }
    vs[pa].sc_temp_delete(vs, vc, pcell);
    vs[pb].sc_temp_add(vs, vc, pcell);
}

void make_nonsplit(vector<State> & vs, vector<Cell> & vc, int pwinner, int ploser, vector<int> & contiguous_cells, bool & cut)
{
    if (!cut)
    {
        vector<int>::iterator iter = contiguous_cells.begin();
        while (iter != contiguous_cells.end())
        {
            int pcell = *iter;
            cell_temp_transfer(vs, vc, ploser, pwinner, pcell);
            if (vs[ploser].cells_contiguous(vs, vc))
            {
                cell_temp_transfer(vs, vc, pwinner, ploser, pcell);
                iter++;
            } else
            {
                cell_temp_transfer(vs, vc, pwinner, ploser, pcell);
                contiguous_cells.erase(iter);
            }
        }
    }

345
if (contiguous_cells.size() == 0)
    cut = false;
else
{
    cerr << "(In the make_nonsplit() function) Exception happens."
    << endl;
    exit(1);
}

// Loser border length
// Calculates any extra number of border length if the passed cell is
// transferred from the loser (between 0 to 6)
int loser_border_length(vector<Cell> & vc, int ploser, int cell)
{
    int length = 0;  // return value
    for (int i = 0; i < 6; i++)
    {
        int neighbor = vc[cell].get_neighbor_cell(i);
        if (neighbor == -1)
            length++;
        if (vc[neighbor].get_which_state() == ploser)
            length++;
    }
    return length;
}

// Winner border length
// Calculates any extra number of border length if the passed cell is
// transferred to the winner (between 0 to 6)
int winner_border_length(vector<Cell> & vc, int pwinner, int cell)
{
    int length = 0;  // return value
    for (int i = 0; i < 6; i++)
    {
        int neighbor = vc[cell].get_neighbor_cell(i);
        if (neighbor == -1)
            length++;
        if (vc[neighbor].get_which_state() != pwinner)
            length++;
    }
    return length;
}

// Shortest loser border length
// Makes a list of cells that make the loser have shortest border
// from a list of contiguous cells (contiguous_cells)
void make_loser_compact(int ploser, vector<Cell> & vc,
                        vector<int> & contiguous_cells)
{
    int each_length;
    int shortest_length = 999;
    for (int i = 0; i < contiguous_cells.size(); i++)
each_length = loser_border_length
    (vc, ploser, contiguous_cells.at(i));
if (shortest_length > each_length)
    shortest_length = each_length;
}
vector<int>::iterator iter = contiguous_cells.begin();
while(iter != contiguous_cells.end())
{
    each_length = loser_border_length(vc, ploser, *iter);
    // Leave the cells that make the loser have shortest border
    if (each_length == shortest_length)
    {
        iter++;
        continue;
    }
else
        contiguous_cells.erase(iter);
}

void make_winner_compact(int pwinner, vector<Cell> & vc,
vector<int> & contiguous_cells)
{
    int each_length;
    int shortest_length = 999;
    for(int i = 0; i < contiguous_cells.size(); i++)
    {
        each_length = winner_border_length
            (vc, pwinner, contiguous_cells.at(i));
        if (shortest_length > each_length)
            shortest_length = each_length;
    }
vector<int>::iterator iter = contiguous_cells.begin();
while(iter != contiguous_cells.end())
{
    // Leave the cells that make the loser have shortest border
    each_length = winner_border_length
        (vc, pwinner, *iter);
    if (each_length == shortest_length)
    {
        iter++;
        continue;
    }
else
        contiguous_cells.erase(iter);
}

void transfer_first_cell(vector<int> & cell_list, vector<State> & vs,
vector<Cell> & vc, int ploser, int claimant, int & how_many)
{
    vector<int>::iterator iter = cell_list.begin();
}
```c
int pcell = *iter;
cell_transfer2(vs, vc, ploser, claimant, pcell);
how_many--;

// Cell-state connection resetting
for (int j = 0; j < vs.size(); j++)
for (int k = 0; k < vs[j].get_sc_size(); k++)
{
  int pcell = vs[j].get_sc(k);
  vc[pcell].set_which_state(vs[j].get_id());
}
for (int m = 0; m < vs.size(); m++)
{
  vs[m].set_snc(vc);
  vs[m].set_sns(vc);
}

// Generate a new state and transfer the first cell of cell_vector
// from the loser to the new state
void new_state_transfer(int new_state_id, vector<int> cell_vector,
  vector<State> & vs, vector<Cell> & vc, int ploser,
  vector<int> & pool_new_states, int & how_many,
  vector<int> & cell_list, ostreams fout, float META_SD)
{
  int pcell = cell_vector[0];
  vs[ploser].sc_delete(vs, vc, pcell);
  State state(new_state_id, pcell, vc, vs);
  float tempa = nr(1.0, META_SD);
  float tempb = nr(1.0, META_SD);
  state.set_state_ura(tempa);
  state.set_state_urb(tempb);
  vs.push_back(state);
  how_many--;
  vector<int>::iterator iter;
  iter = find(cell_list.begin(), cell_list.end(), pcell);
  cell_list.erase(iter);
  int ps = vs.back().get_id();
  pool_new_states.push_back(ps);
  print_new_state(fout, vs, vc, ploser, ps, pcell);
}

// Generate a new state and transfer the parameter cell
// from the loser to the new state
void new_state_transfer_cell (int new_state_id, int pcell, vector<State> & vs,
  vector<Cell> & vc, int ploser, vector<int> & pool_new_states, int &
  how_many, vector<int> & cell_list, ostream& fout, float META_SD)
{
  vs[ploser].sc_delete(vs, vc, pcell);
  State state(new_state_id, pcell, vc, vs);
  float tempa = nr(1.0, META_SD);
  float tempb = nr(1.0, META_SD);
  state.set_state_ura(tempa);
  state.set_state_urb(tempb);
  vs.push_back(state);
}
```
how_many--; 
vector<int>::iterator iter; 
iter = find(cell_list.begin(), cell_list.end(), pcell); 
cell_list.erase(iter); 
int ps = vs.back().get_id(); 
pool_new_states.push_back(ps); 
print_new_state(fout, vs, vc, ploser, ps, pcell);

// // Cell-state connection resetting
// // Generate a new state and transfer the cell passed as a parameter (pcell)
// // from the loser to the new state; Unlike new_state_transfer() function,
// // this function does not use pool_new_states vector
// // (Only for the fragmented, contiguous cell vectors ready for new states)

void new_state_transfer2(int new_state_id, int pcell, vector<Cell> & vs,
                        vector<Cell> & vc, int ploser, float META_SD)
{
    vs[ploser].sc_delete(vs, vc, pcell); 
    State state(new_state_id, pcell, vc, vs); 
    float tempa = nr(1.0, META_SD); 
    float tempb = nr(1.0, META_SD); 
    state.set_state_ura(tempa); 
    state.set_state_urb(tempb); 
    vs.push_back(state); 
    int ps = vs.back().get_id(); 

    // // Cell-state connection resetting

    for (int j = 0; j < vs.size(); j++) 
        for (int k = 0; k < vs[j].get_sc_size(); k++) 
        {
            int pcell = vs[j].get_sc(k); 
            vc[pcell].set_which_state(vs[j].get_id()); 
        }

    for (int m = 0; m < vs.size(); m++) 
        { 
            vs[m].set_snc(vc); 
            vs[m].set_sns(vc); 
        }

    // // Create a vector of contiguous cell blocks from a group of fragmented cells
}
vector< vector<int> > make_contiguous_blocks
(vector<Cell> & vc, vector<int> fragmented)
{
    vector<int> remains;
    vector< vector<int> > v_conti_blocks; // return value
    if (fragmented.size() == 0) return v_conti_blocks;
    for (int i = 0; i < fragmented.size(); i++)
        remains.push_back(fragmented[i]);

    // Make a vector "temp" and search through "fragmented" vector
    // to find any cells that are contiguous to a cell in the "temp"
    while (remains.size() > 0)
    {
        int pcell = remains.back();
        remains.pop_back();
        vector<int> temp;
        temp.push_back(pcell);
        int pre_sentinel = 0; // Default value to make it different
        int post_sentinel = 1; // Default value to make it different
        vector<int>::iterator iter;
        while (pre_sentinel != post_sentinel) // temp.size() changed
        {
            pre_sentinel = remains.size();
            for (int i = 0; i < temp.size(); i++)
            {
                int temp_cell = temp[i];
                for (int j = 0; j < remains.size(); j++)
                {
                    int cell = remains[j];
                    if (vc[temp_cell].is_neighbor_cell(vc, cell))
                    {
                        temp.push_back(cell);
                        iter = find(remains.begin(),
                                    remains.end(), cell);
                        remains.erase(iter);
                    }
                }
            }
            post_sentinel = remains.size();
        }
        // Now temp's all elements are contiguous; now into v_conti_blocks
        v_conti_blocks.push_back(temp);
    }
    return v_conti_blocks;
}

// Choose a block that has most cells from a vector of contiguous cell blocks
vector<int> choose_main_block(vector< vector<int> > v_conti_blocks)
{
    if (v_conti_blocks.size() == 0)
    {
        cerr << "This vector of contiguous blocks does not have any element"
             << ": in choose_main_block()"
             << endl;
        exit(1);
    }
    vector<int> main_block; // return value
    main_block = v_conti_blocks.at(0);
    if (v_conti_blocks.size() > 1)
        for (int i = 1; i < v_conti_blocks.size(); i++)
if (v_conti_blocks[i].sized > main_block.size())
    main_block = v_conti_blocks[i];
return main_block;

// Check whether all cells are contiguous in a vector
//
bool is_cell_vector_contiguous(vector<State> & vs, vector<Cell> & vc,
    vector<int> cell_vector)
{
    if (cell_vector.size() > 1)
    {
        for (int i = 0; i < cell_vector.size(); i++)
        {
            int celli = cell_vector[i];
            vector< vector<int> > contiguous_vv;
            contiguous_vv = vc[celli].set_contiguous_vv(vs, vc);
            if (contiguous_vv.size() != cell_vector.size())
                return false;
        }
        return true;
    }
    else if (cell_vector.size() == 1) return true;
    else
    {
        cerr << "This vector passed to is_cell_vector_contiguous() "
            << "function does not have any cell." << endl;
        exit(1);
    }
}

// Get the number of contiguous blocks in a vector of fragmented cells
//
int get_cell_vector_block_number(vector<State> & vs, vector<Cell> & vc,
    vector<int> cell_vector)
{
    if (cell_vector.size() == 0)
    {
        cerr << "This vector passed to get_cell_vector_block() "
            << "function does not have any cell." << endl;
        exit(1);
    }
    if (!is_cell_vector_contiguous(vs, vc, cell_vector)) return 1;
    vector<int> remains;
    vector< vector<int> > v_conti_blocks; // its size will be returned
    for (int i = 0; i < cell_vector.size(); i++)
    {
        remains.push_back(cell_vector[i]);
    }
    while (remains.size() > 0)
    {
        int pc = remains.back();
        remains.pop_back();
        if (v_conti_blocks.size() == 0)
        {
            vector<int> temp;
            temp.push_back(pc);
            v_conti_blocks.push_back(temp);
            continue; // Go back to while loop
        }
    }
}
int sentinel = 0;
for (int a = 0; a < v_conti_blocks.size(); a++)
{
    vector<int> temp = v_conti_blocks[a];
    for (int b = 0; b < temp.size(); b++)
    {
        int pcref = temp[b];
        if (vc[pcref].is_neighbor_cell(vc, pc))
        {
            temp.push_back(pc);
            sentinel = 1;
            break;
        }
    }
    if (sentinel == 1) continue;
}
// If the cell is not contiguous to any cells
vector<int> another;
another.push_back(pc);
v_conti_blocks.push_back(another);
} // End of while
return v_conti_blocks.size();