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. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

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.

UMI
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700  800/521-0600
SOLUTION AND ANALYSIS FOR ROUTING IN
POINT-TO-POINT FREIGHT DELIVERY SYSTEMS
WITH BREAKBULKS

DISSERTATION

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

By

Joseph T. Chao, B.S., M.S.

The Ohio State University

1996

Dissertation Committee:
Professor Marc E. Posner, Adviser
Professor Rakesh V. Vohra
Professor Nicholas G. Hall

Approved by

Adviser
Industrial and Systems
Engineering Graduate
Program
ABSTRACT

Point-to-point delivery systems ship freight over networks where there are many origins and many destinations. Shipping directly from origins to destinations is often too costly when the freight size is less than the vehicle capacity. This is frequently the case for less-than-truck-load carriers. These carriers consolidate smaller freight flows into larger flows at breakbulk terminals.

This research studies and provides polynomial time procedures for various freight delivery systems with single or multiple breakbulk terminals. We model the shipping cost as the cost of dispatching and sending a trailer along a given arc. Thus, the cost on each arc depends only on the number of trailers traveling on the arc. We assume that all trailers have the same capacity. The problem with one origin and one breakbulk is shown to be binary $NP$-hard. We also establish that more general problems are unary $NP$-hard. The heuristics proposed in this research are simple and easy to implement.
To My Parents and My Wife.
ACKNOWLEDGMENTS

My sincerest gratitude and appreciation are expressed to my adviser Dr. Marc E. Posner for his continuous guidance and encouragement throughout this research. He has given generously his time and has motivated me to complete this work. In every respect, he has been an extremely helpful adviser and mentor. Without his enthusiasm and vision, this task would have been impossible. Appreciation is extended to the other members of my committee, Dr. Rakesh V. Vohra and Dr. Nicholas G. Hall, for their valuable comments and suggestions.

I am grateful to the Department of Industrial and Systems Engineering for providing an excellent learning environment and thorough career preparation, which enabled me to accomplish my goal at The Ohio State University. Special thanks go to the manager of ISE Computer and Control Lab and my best friend, Cedric Sze, for his support and encouragement over many years. Very special appreciation goes to Mr. Ty Jenkins, CEO of Lynden Corporation, for his long time support and understanding. I am extremely grateful to him for letting me leave for school on several occasions that were critical to the business.

Finally, I wish to express my heartfelt thanks to my beautiful wife for her infinite love, patience and sacrifice throughout this seemingly never-ending pursuit of my dreams.
VITA

December 20, 1957 .......................... Born - Taipei, Taiwan, R.O.C.

1980 ................................. B.S. Applied Mathematics,
                Chung-Yuan University,
                Chung-Li, Taiwan, R.O.C.

1986 ................................. M.S. Operations Research,
                Case Western Reserves University,
                Cleveland, Ohio.

1986-1988 ............................. Graduate Teaching Associate,
                The Ohio State University,
                Columbus, Ohio.

1989-1992 ............................. Computer and Control Lab Assistant,
                Industrial and Systems Engineering,
                The Ohio State University.

1992-1993 ............................. Engineering Specialist,
                ML Technologies,
                Idaho Falls, Idaho.

1994-present ........................ Director, Software Development,
                Lynden Corporation,
                Idaho Falls, Idaho.

PUBLICATIONS

Research Publications

Co, Henry, J. Chao and S. Chen, “Sequencing in Flexible Manufacturing Systems and
FIELDS OF STUDY

Major Field: Industrial and Systems Engineering

Studies in:

Operations Research  Prof. Marc E. Posner
Automation            Prof. Gary Maul
Database Systems      Prof. Dik Lee
# TABLE OF CONTENTS

Abstract .................................................. ii
Dedication .............................................. iii
Acknowledgments ....................................... iv
Vita ......................................................... v
List of Tables .......................................... ix
List of Figures .......................................... x

Chapters:

1. Introduction ........................................... 1

2. Literature Review ...................................... 5

3. Single Origin, Single Breakbulk Problem .............. 10
   3.1 The Complexity of $R$ ............................... 13
   3.2 Fully Polynomial-Time Approximation Scheme ........ 15
   3.3 A Class of Heuristic Procedures .................... 17
   3.4 Conclusion .......................................... 30

4. General Single Breakbulk Problem .................... 32
   4.1 Notation and Definitions ......................... 33
   4.2 The Complexity of $R_s$ ............................ 35

vii
4.3 A Heuristic Procedure ........................................ 38
4.4 Conclusion ....................................................... 46

5. Clique Breakbulk Problem ....................................... 47
  5.1 Problem Formulation and Notation .......................... 48
  5.2 A Heuristic Procedure ....................................... 49
  5.3 Conclusion ................................................... 59

6. Multiple Breakbulk Problem ..................................... 61
  6.1 Problem Formulation and Notation .......................... 61
  6.2 A Heuristic Procedure ....................................... 63
  6.3 Conclusion ................................................... 71

7. Summary and Future Work ....................................... 72

Appendices:

A. Comparison of H1 and $\mathcal{H}(g)$ with $g_j = a_j + \Delta$ ............... 73

B. Empirical study of H1, H2 and the optimal routing ......................... 80

C. Empirical study for Procedure HM ................................ 89

References ........................................................ 109


### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Worst-case example for H1.</td>
<td>21</td>
</tr>
<tr>
<td>A.1 $t$-values of $z^1 - z^g$ with $g_j = a_j + \Delta$ for 10,000 replications.</td>
<td>74</td>
</tr>
<tr>
<td>A.2 $t$-values of $z^1 - z^g$ with $g_j = a_j + \Delta$ for 100,000 replications.</td>
<td>75</td>
</tr>
<tr>
<td>B.1 Average differences between $z^1$, $z^2$, $z^*$ and $z^{LP}$.</td>
<td>81</td>
</tr>
<tr>
<td>C.1 Average differences of $z^h$ and $z^{LP}$.</td>
<td>91</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Single origin, single breakbulk delivery network</td>
</tr>
<tr>
<td>4.1</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>General single breakbulk delivery network</td>
</tr>
<tr>
<td>5.1</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>Clique breakbulk delivery network with 4 breakbulks</td>
</tr>
<tr>
<td>6.1</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>Worst-case example for $R_m$</td>
</tr>
<tr>
<td>C.1</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>Network of $R_m$ with four and six breakbulks studied.</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

Point-to-point delivery systems ship freight over networks where there are many origins and many destinations. Each item of freight has its own origin and destination which forms an origin-destination pair. Terminals on the network can be classified into two types: end-of-line terminals and breakbulk terminals. End-of-line terminals are origins and/or destinations for freight while breakbulks are consolidation or intermediate transshipment points. Shipping directly from origins to destinations is often too costly when the freight size is less than the vehicle capacity, less-than-truck-load (LTL). Freight consolidation combines smaller freight flows into larger flows. At each breakbulk, freight flows from origins or other breakbulks are unloaded, sorted by destinations, and then reloaded into outbound vehicles for further dispatch. By consolidation, carriers may reduce shipping costs and achieve other service advantages. However, with full or nearly full loads, it may be less expensive to ship the freight directly to its destination. The freight delivery problem determines how each item should be routed to minimize the total transportation cost. An in-depth introduction to the LTL carriers, consolidation processes, and other carrier operations is given in Taff (1986).
The freight delivery problem has drawn considerable attention from researchers, especially after the deregulation act of motor carriers in 1980. The deregulation made entry easier, eliminated some route and rate restrictions. A review and analysis of The Motor Carrier Act of 1980 is given in Harper (1980). In recent years, freight routing and consolidating has become even more important because a large volume of freight with smaller item sizes are generated by the increasingly competitive and demanding economy. For many carriers, a one percent cost savings can mean millions of dollars in profit per year. However, despite of the importance and high payoff, optimal solutions have not been obtained for most real world applications of freight delivery problems. Difficulties arise because of the complexity and the large size of the problems.

The goal of this research is to develop an understanding of some of the basic structures of LTL problems. We study and provide solution procedures for various freight delivery systems with breakbulks. In our models, the cost of shipping on an arc is the cost of dispatching and sending a trailer along the arc. Thus, the shipping cost on each arc depends only on the number of trailers traveling on the arc and is independent of the freight size. We assume that all trailers have the same capacity. Also, an individual item may be split and shipped through different routes. Our problem is given a set of items to be delivered through a network with sets of origin, destination and breakbulk terminals, and the shipping cost per trailer for each arc, find the routing strategy that minimizes the total shipping costs.

In the next chapter, we discuss the significance of this research and present a survey of the literature for various approaches to solving freight delivery problems.

Our analysis first considers a basic single breakbulk delivery system. We then
study more general systems with multiple breakbulks. Chapter 3 provides solutions and analyses for the freight delivery problem where there is a single breakbulk, a single origin, and many destinations. We show that the recognition version of the problem is binary \( \mathcal{NP} \)-complete. A fully polynomial-time approximation scheme is developed (for a definition, see Papadimitriou and Steiglitz, 1982). We then propose a class of heuristic procedures, \( H(g) \). Each procedure in this class uses a specific function \( g \) to evaluate the shipping cost and determine the routing strategy for each item. When \( g \) can be computed in polynomial time, \( H(g) \) is a polynomial time procedure. We find a procedure, called H1, which has the best worst-case error bound among all possible heuristics in \( H(g(a)) \), where \( a \) is the vector of item sizes. The error bound of H1 goes to zero as the number of items goes to infinity provided that some problem parameters are bounded. An improved polynomial time heuristic over H1, called H2, is then proposed. An empirical study shows that H2 produces near optimal solutions. However, unlike H1 which uses only the information pertaining to individual item sizes, H2 uses all of the problem information when determining a route for each item.

In Chapter 4, we study the routing strategies for the single breakbulk problem where there are many origins and many destinations. The recognition version of the problem is shown to be unary \( \mathcal{NP} \)-complete. Also, a polynomial time heuristic procedure is proposed which finds a solution value with relative error of one (the solution value is no more than twice the optimal value).

For problems with more than one breakbulk, Chapter 5 studies the freight delivery problem where every pair of breakbulk terminals are connected with arcs (clique breakbulks). We propose a heuristic procedure that has a relative error of two.

In Chapter 6, we study a general freight delivery system with multiple breakbulks
that do not necessarily form a clique. We develop a heuristic that first converts the
general problem to the problem with clique breakbulks. Then, we use the procedure
in Chapter 5 to solve the problem. Using this method, we solve the general problem
with a relative error of \((b - 1)(b - 2) + 1\), where \(b\) is the number of breakbulks. An
empirical study shows that the heuristic performs well in general.
CHAPTER 2

LITERATURE REVIEW

There is considerable research examining freight consolidation and delivery problems. To study the effect of freight consolidation, Daganzo (1987) demonstrates that cost can be reduced in logistic networks by allowing for breakbulks. Also, he shows how to obtain a near optimal number of breakbulks and how to create an approximate network plan. Hall (1987) compares four strategies for routing shipments through breakbulks over a rectangular grid. He uses the number of transportation links and the number of breakbulks as the measurement of consolidation to evaluate the strategies.

Some empirical studies use computer-based simulation models. Masters (1980) develops a heuristic to select breakbulks and studies the consolidation effects on customer service in terms of holding and delivery times using a discrete simulation model. Jackson (1981) simulates a distribution system and evaluates some freight consolidation strategies. Ha et al. (1988) study the impact of freight consolidation on a logistic system. All of these empirical studies show that freight consolidation can reduce transportation costs.

The location of the breakbulks play an important role in the effectiveness of delivery systems. Researchers such as Hall (1985), O'Kelly (1986) and Iyer and Ratliff
(1990) formulate the distribution problem as a facility location problem. They obtain the optimal breakbulk locations under certain constraints. Often, breakbulks are already in place and difficult to change. As a result, many researchers have studied the routing strategies that ship freight over a large network. Early approaches to the freight delivery problem use heuristic rules. Temple et al. (1978) develop heuristics that apply a predetermined set of rules to calculate if there is sufficient flow to have a direct shipment between any two terminals. Barker et al. (1981) present a heuristic that simulates the movements of freight flows through the network. They first obtain an initial feasible solution based on rules. Starting with this solution, the model analyzes all origin-destination pairs and suggests possible consolidations. The heuristic also checks for items scheduled through breakbulks and determines whether direct shipping might be an improvement. Other approaches to the freight delivery problem develop optimization models. Roy and Delorme (1989) present a network planning system using the modeling and algorithmic framework developed in Crainic and Rousseau (1985). The freight delivery problem is formulated as a nonlinear mixed integer programming problem where service frequency and freight volume for each route are explicit decision variables. Intercity transportation time, vehicle waiting time, and freight unloading, sorting and reloading times are estimated to model the service penalty costs and capture the congestion phenomena at breakbulks. Two subproblems, service network design, and traffic routing and distribution, are solved alternately until a satisfactory solution is reached.

Assuming piecewise linear concave shipping cost functions on each arc, Klincewicz (1990) uses facility location techniques to solve freight a delivery problem. He shows
that if either the source-to-breakbulk or the breakbulk-to-destination shipping cost in
the delivery problem is linear, the problem can be solved optimally by decomposing
the problem into a set of uncapacitated facility location problems. For the more gen-
eral piecewise linear concave cost delivery problems, heuristic methods are proposed
that solve a sequence of linear problems.

The freight delivery problem is a type of network design problem, which can be
formulated as an integer programming problem. Magnanti and Wong (1984) discuss
network design models and algorithms in general. They also describe a wide variety
of transportation planning applications and provide comprehensive reviews of some
approaches using integer programming.

To solve the network design problem, Balakrishnan (1984) uses a Lagrangian-
based algorithm that includes heuristic and lower bounding procedures as well as a
problem reduction method to identify near optimal solutions. Lamar and Sheffi (1987)
present an implicit enumeration procedure to determine a lower bound and a near op-
timal solution. Balakrishnan et al. (1989) develop a generalized family of dual-ascent
algorithms for a class of uncapacitated network design problems. Powell and Sheffi
(1989) design and implement an interactive optimization system. They formulate a
large mixed integer programming problem, and use a decomposition scheme based
on the problem structure. Once a solution is obtained from the optimization model,
local improvement heuristics are used to suggest network design improvements. Then
based on the suggested improvements, interactive decisions can be made regarding
certain complex constraints and tradeoffs. Bertsimas et al. (1995) use randomize
rounding algorithms with dependency in the rounding process to produce approxi-
mation bounds for several problems including the network design problem. For the
network design problem, an error bound of \( \max_i \{ [a_i/C]C/a_i \} \) is obtained where \( a_i \) is the size of item \( i \) and \( C \) is the vehicle capacity. The bound is two when \( a_i \geq C \) for all \( i \).

For point-to-point delivery systems, Leung et al. (1990) formulate the problem as a nonlinear mixed integer program. The problem is decomposed into an assignment problem with capacity constraints and a routing problem between breakbulks. The assignment problem assigns each origin-destination pair to a first and a last breakbulk. Then, the routing problem finds a minimum cost routing for the assigned items among breakbulks. A solution to the routing problem is found by using Lagrangian relaxation. The assignment and routing problems are iteratively solved until a satisfactory solution is reached. Li et al. (1992) study the point-to-point delivery problems where the capacity of a trailer is a fixed number of items, and that exactly one item is sent from each origin and exactly one item is shipped to each destination. They show that the recognition versions of all point-to-point delivery problems where the trailer holds at least two items are unary \( \mathcal{NP} \)-complete. Optimal algorithms are also developed for some special cases.

Studying a portion of the freight delivery system, Barnhart and Kim (1995) model a regional LTL operations as an integer network. They consider a single breakbulk with a set of EOL terminals. There are a set of tractors that moves trailers between EOL terminals and the breakbulk. The tractors start and end at the breakbulk. The objective is to determine tractor pickup and delivery routes to minimize the cost. Procedures are developed to solve the problem using minimum weight matching algorithm. Computational results show that near-optimal solutions are obtained.

All of the above optimization models formulate the various freight delivery systems
as large scale programming problems. However, because of the complexity of the models and solution procedures, performance is only measured by empirical testing. Furthermore, each model has its own special set of assumptions which are not easy to modify.
CHAPTER 3

SINGLE ORIGIN, SINGLE BREAKBULK PROBLEM

In many distribution systems, goods are shipped from one source to many destinations. For example, most firms and industries distribute their products from a manufacturing facility to their customers. If the firms ship their products directly from source to destination, then they often encounter a large number of relatively small size (less-than-truck-load) shipments. In moving these small shipments, the firms either pay higher rates to common carriers or have large unused capacity in their trucks. Therefore, freight consolidation, which combines small items into larger shipments, is often used. In freight consolidation, customers in a given region are typically assigned to a breakbulk terminal (a decentralized distribution center or a consolidation point). Orders from customers in the same region can be combined and shipped together to the breakbulk in the region. At the breakbulk, combined shipments are unloaded, sorted by destination, and then reloaded into outbound vehicles and shipped to customers.

An example of a single breakbulk system is when there is one manufacturing plant and one regional warehouse. Goods can be shipped to the customer either from the plant or the warehouse. This model is also useful when considering expansion or contraction of an existing system. The analysis provides a upper bound on the cost
of adding or deleting a destination to an existing system. Since the one origin and many destination system is equivalent to a many origin and one destination system, this work can also be used to evaluate the systems where many sources supply goods for a single manufacturing plant.

Consider the freight delivery network as shown in Figure 3.1 that has one breakbulk terminal \( b \), one origin (supply) terminal \( s \), and destination terminals \( N = \{1, 2, \ldots, n\} \). Item \( j \in N \) has \( a_j \) trailer-loads to be delivered from origin \( s \) to destination \( j \) through a set of directed arcs (direct shipping routes). Associated with each arc \((s, j), (s, b)\) and \((b, j)\) is the non-negative cost per trailer \( c_{sj}, c_{sb} \) and \( c_{bj} \), respectively. The objective of the problem is to find a routing that minimizes the total shipping costs. When \( c_{sj} \leq c_{bj} \), shipping item \( j \in N \) directly is no more costly than going through the consolidation process at the breakbulk. Consequently, to obtain the lowest shipping cost, item \( j \) is shipped directly to destination terminal \( j \). Therefore, without loss of generality, we assume \( c_{sj} > c_{bj} \) for \( j \in N \).

This research assumes that each item is separable and the pieces can be shipped through different routes. However, for each item \( j \in N \) where \( a_j < 1 \), it is no more costly to ship the entire item direct than it is to ship part of the item direct and the remaining part of the item through the breakbulk. Consequently, it is never advantageous to split an item. Furthermore, a full trailer load has no need for consolidation, and it is at least as expensive to split a full trailer load of items into different routes. Thus, for each item \( j \in N \), if \( a_j \geq 1 \), then \( \lfloor a_j \rfloor \) full trailers are dispatched, and only the fractional part of the item is considered for consolidation. Both optimal and heuristic routings ship the full trailer loads direct if \( c_{sj} \leq c_{sb} + c_{bj} \); otherwise, they are shipped through the breakbulk without consolidation. As a result, we assume that
Figure 3.1: Single origin, single breakbulk delivery network

\[ 0 < a_j < 1 \text{ for } j \in N. \]

We define the decision variables

\[ x_j = \begin{cases} 
1 & \text{if item } j \text{ goes from } b \text{ to } j, \\
0 & \text{otherwise}, 
\end{cases} \quad j \in N, \]

\[ y = \text{the number of trailers required from } s \text{ to } b. \]

The freight delivery problem can be stated as

\[ \text{(R)} \]

\[
\begin{align*}
\text{Minimize} & \quad z(x, y) = \sum_{j=1}^{n} c_{x_j}(1 - x_j) + c_{xy}y + \sum_{j=1}^{n} c_{x_j} x_j \\
\text{subject to} & \quad \sum_{j=1}^{n} a_j x_j - y \leq 0 \\
& \quad x_j \in \{0, 1\}, \quad j \in N \\
& \quad y \in \{0, 1, 2, \ldots, \lceil \sum_{j=1}^{n} a_j \rceil \}. 
\end{align*}
\]
The more restrictive problem where each item must be shipped entirely on one route (both full and partial trailer loads) can be transformed into our problem where the full and partial trailer loads may be shipped on different routes. Define
\[ a'_j = a_j - \lfloor a_j \rfloor. \]

Since full trailer loads do not affect consolidation, we assign the full trailer cost to given routes. Let
\[ c'_{aj} = c_{aj} + \lfloor a_j \rfloor c_{aj}, \]
\[ c'_{bj} = c_{bj} + \lfloor a_j \rfloor (c_{ab} + c_{bk}). \]

Then, the freight delivery problem with parameter \( c_{ab}, c'_{aj}, c'_{bj} \) and \( a'_j \) for \( j \in N \) is an instance of the problem \( R \) where \( a'_j < 1 \) for \( j \in N \). The transformation from the restricted problem to the separable problem extends in a natural way to each of the problems discussed in this research.

Throughout the research, we denote \( z^* \) and \( z^h \) as the optimal and heuristic solution values, respectively. Also, define \( |S| \) to be the cardinality of set \( S \). If \( S \subseteq N \), then \( \bar{S} = N \setminus S \).

### 3.1 The Complexity of R

We determine the complexity of \( R \) by a polynomial time reduction from the following binary \( \mathcal{NP} \)-complete problem.

**0-1 Knapsack Problem (KS).** Given integers \( q_1, q_2, \ldots, q_m \) and \( d \), does there exist a set \( Q \subseteq \{1, 2, \ldots, m\} \) such that \( \sum_{j \in Q} q_j = d \)?
Theorem 3.1 The recognition version of $R$ is binary $\mathcal{NP}$-complete.

Proof. The recognition version of $R$ is in $\mathcal{NP}$ because each yes instance can be verified in linear time. Given an instance of KS, consider the instance of $R$ where

\[
\begin{align*}
n &= m + 1, \\
a_j &= \frac{q_j}{d+1}, & j = 1, 2, \ldots, m, \\
a_{m+1} &= \frac{1}{d+1}, \\
c_{aj} &= \frac{q_j}{d+1}, & j = 1, 2, \ldots, m, \\
c_{a_{m+1}} &= 2, \\
c_{sb} &= 2, \\
c_{bj} &= 0, & j = 1, 2, \ldots, m + 1, \\
K &= 2 + \frac{1}{d+1}(\sum_{j=1}^{m} q_j - d).
\end{align*}
\]

We show that there exists a solution to KS if and only if there is a solution to this instance of $R$ with cost no larger than $K$.

$(\implies)$ If there exists a subset $Q$ such that $\sum_{j \in Q} q_j = d$, then

\[
a_{m+1} + \sum_{j \in Q} a_j = \frac{1}{d+1} + \frac{d}{d+1} = 1.
\]

The routing that ships all items $j \in Q \cup \{m + 1\}$ through the breakbulk $b$ and the remainder of the items directly to the destination $j$ has cost

\[
c_{sb} + \sum_{j \in Q} c_{aj} = 2 + \sum_{j \in Q} \frac{q_j}{d+1}
= 2 + \frac{1}{d+1} \sum_{j=1}^{m} q_j - \frac{d}{d+1}
= K.
\]

$(\impliedby)$ Suppose there is a solution to the instance of $R$ with cost no larger than $K$. If item $m + 1$ is shipped direct, then the cost of the solution is at least $2 + \sum_{j=1}^{m} \frac{q_j}{d+1} > K$. Consequently, item $m + 1$ must go through the breakbulk.
Let \( Q \subseteq \{1,2,\ldots,m\} \) be the set of items that are also shipped through the breakbulk. Then, the cost is

\[
z = 2 \left[ \frac{1}{d+1} + \sum_{j \in Q} \frac{q_j}{d+1} \right] + \sum_{j \in Q} \frac{q_j}{d+1}
\]

\[
= 2 \left[ \frac{1}{d+1} \left( 1 + \sum_{j \in Q} q_j \right) \right] + \frac{1}{d+1} \left( \sum_{j=1}^m q_j - \sum_{j \in Q} q_j \right).
\]

If \( \sum_{j \in Q} q_j < d \), then

\[
z = 2 + \frac{1}{d+1} \left( \sum_{j=1}^m q_j - \sum_{j \in Q} q_j \right)
\]

\[
> 2 + \frac{1}{d+1} \left( \sum_{j=1}^m q_j - d \right) = K.
\]

If \( \sum_{j \in Q} q_j > d \), then

\[
z \geq 2 \left[ \frac{1}{d+1} \left( 1 + \sum_{j \in Q} q_j \right) \right] + \frac{1}{d+1} \sum_{j=1}^m q_j - \left[ \frac{1}{d+1} \sum_{j \in Q} q_j \right]
\]

\[
\geq 2 + \frac{1}{d+1} \sum_{j=1}^m q_j > K.
\]

Hence, if the solution to R has cost no more than \( 2 + \frac{1}{d+1} (\sum_{j=1}^m q_j - d) \), then there must be a set \( Q \subseteq N \) such that \( \sum_{j \in Q} q_j = d \). The set provides a solution to KS.  

**3.2 Fully Polynomial-Time Approximation Scheme**

We develop a fully polynomial-time approximation scheme (FPTAS) that produces a heuristic solution for problem R with a cost which is within a prespecified relative error \( \varepsilon \) (\( \varepsilon \)-approximate solution). The FPTAS has a time bound that is polynomial in the length of the problem instance and \( 1/\varepsilon \).

Although \( \mathcal{NP} \)-hard, an optimal solution for problem R can be found by a pseudo-polynomial time algorithm. For any fixed number of trailers, \( v \), on arc \((s,b)\), R becomes the following Knapsack Problem:
KS\(_v\)  
\begin{align*}
\text{Minimize} & \quad z_v(x) = \sum_{i=1}^n c_i x_i \\
\text{subject to} & \quad \sum_{i=1}^n a_i x_i \leq v \\
& \quad x_i \in \{0, 1\}, \quad i \in N.
\end{align*}

where \(c_i = c_{si} - c_{bi}\) for \(i \in N\), and \(v \in \{1, 2, \ldots, \lceil \sum_{i=1}^n a_i \rceil \}\). Therefore, we can use the dynamic programming procedure, DP-III, by Papadimitriou and Steiglitz (1982) to solve KS\(_v\) in \(O(n^2 z_v^*)\) time, where \(z_v^*\) is the optimal solution value. Furthermore, for any \(\varepsilon\), we can design an \(\varepsilon\)-approximate algorithm that finds a solution to KS\(_v\) with cost that is no larger than \((1+\varepsilon)z_v^*\). Let \(c_{max} = \max_{i \in N} \{c_i\}\). Because \(z_v^* \leq nc_{max}\), the time required by DP-III is \(O(n^3 c_{max})\). By truncating the last \(l = \lfloor \log(\varepsilon c_{max}/n) \rfloor\) digits of \(c_i\), \(i = 1, 2, \ldots, n\), and then applying DP-III to the truncated problem, we have an \(\varepsilon\)-approximate algorithm for KS\(_v\) with \(\varepsilon = n10^l/c_{max}\). For the truncated problem, the time required becomes \(O(n^3 c_{max}10^{-l})\) because the \(c_i\)'s have been divided by \(10^l\). Hence, the time complexity of the \(\varepsilon\)-approximate algorithm is \(O(n^4/\varepsilon)\).

Because \(v \leq V \leq n\), where \(V = \lceil \sum_{i=1}^n a_i \rceil\), if we repeat the \(\varepsilon\)-approximate algorithm for KS\(_v\) for \(v = 1, 2, \ldots, V\), then we have an FPTAS for R. A formal description of the FPTAS for R is now presented.

**Procedure FPP**

0. Let \(c_{max} = \max_{i \in N} \{c_{si} - c_{bi}\}\), \(V = \lceil \sum_{i=1}^n a_i \rceil\), \(l = \lfloor \log(\varepsilon c_{max}/n) \rfloor\) and \(c_i = \lfloor (c_{si} - c_{bi})/10^l \rfloor 10^l\) for \(i = 1, 2, \ldots, n\).

1. For \(v = 1, 2, \ldots, V\),
   
   (a) Apply DP-III to the instance of KS\(_v\), and obtain an optimal solution \((x_1^v, x_2^v, \ldots, x_n^v)\).
   
   (b) Let \(S_v = \{i \in N \mid x_i^v = 1\}\), and \(z(v) = \sum_{i=1}^n c_{si} - \sum_{i \in S_v} c_i + c_{ab}v\).
2. Find $v^* = \min_{v \in V} \{ z(v) \}$. Ship all items in $S_{v^*}$ through the breakbulk $b$, and other items directly to destination $i \in N$. The cost of this routing is $z_{v^*}(v^*)$.

The dynamic programming procedure in Step 1 takes $O(n^4/\varepsilon)$ time to complete and we repeat the procedure $V < n$ times. Consequently, the time complexity for Step 1 is $O(n^5/\varepsilon)$. Each of the other steps require only $O(n)$ time. Therefore, procedure FPP is an FPTAS which produces as $\varepsilon$-approximate solution for $R$ with a time bound of $O(n^5/\varepsilon)$.

### 3.3 A Class of Heuristic Procedures

We propose a class of polynomial time heuristic procedures to solve $R$. The heuristic uses a function to evaluate the shipping cost and determine the routing strategy for each individual item. For each distinct function, there is a corresponding heuristic.

For ease of notation, let

$$g_j \equiv g_j(a_1, a_2, \ldots, a_n).$$

Define $G = \{ j \in N \mid c_{sj} \leq c_{bj} + g_jc_{sb} \}$ where $g_j : \mathbb{R}^n \to \mathbb{R}$. For a specified function $g$, consider the following heuristic procedure that ships each item $j \in G$ directly to its destination and ships all other items through the breakbulk.

**Procedure $\mathcal{H}(g)$**

1. Given $g_j$ for $j \in N$, let $G = \{ j \in N \mid c_{sj} \leq c_{bj} + g_jc_{sb} \}$.

2. Let

$$z^g = \min \left\{ \sum_{j=1}^{n} c_{sj}, \sum_{j=1}^{n} c_{bj} + \left[ \sum_{j=1}^{n} a_j \right] c_{sb}, \sum_{j \in G} c_{sj} + \sum_{j \in G} c_{bj} + \left[ \sum_{j \in G} a_j \right] c_{sb} \right\}.$$

Ship all items directly to the destinations if $z^g = \sum_{j=1}^{n} c_{sj}$. 

17
Otherwise, ship all items through \( b \) if 
\[
z^g = \sum_{j=1}^{n} c_{bj} + [\sum_{j=1}^{n} a_j] c_{sb}. \]

Otherwise, ship each item \( j \in G \) directly to \( j \), and all other items through \( b \).

The cost is \( z^g \). Stop.

Procedure \( \mathcal{H}(g) \) is a class of heuristics that uses the function \( g \) to evaluate the shipping cost of each item. Step 1 specifies a set of items \( G \) for which shipping the items directly to their destinations is likely to be cheaper. Step 2 determines whether to ship all of the items directly to their destinations, to ship them all through the breakbulk, or to ship them based on \( G \).

Note that when \( n \leq 2 \), \( \mathcal{H}(g) \) produces an optimal solution. Suppose an instance of \( R \) has two items. If the optimal routing ships them both direct or through the breakbulk, then \( z^g \) is optimal. If the optimal routing ships item 1 direct and item 2 through \( b \), then \( c_{b2} > c_{sb} + c_{b2} \) and item 2 is not in \( G \). Consequently, \( \mathcal{H}(g) \) also ships item 2 through \( b \), and gives an optimal solution.

To compute the time complexity, Step 2 requires \( O(n) \) time. Hence, \( \mathcal{H}(g) \) solves \( R \) in \( O(n) \) if \( g_1, g_2, \ldots, g_n \) in Step 1 can be obtained in \( O(n) \) time. Thus, \( \mathcal{H}(g) \) is a polynomial time procedure if \( g_1, g_2, \ldots, g_n \) can be calculated in polynomial time.

The performance of \( \mathcal{H}(g) \) is directly related to the selection of \( g \). A specific example of \( \mathcal{H}(g) \), \( H_1 \), is presented for analysis. Let \( H_1 \equiv \mathcal{H}(g^1) \) where

\[
g^1_j = a_j \text{ for all } j,
\]

and

\[
G_1 = \{ j \in N \mid c_{sj} \leq c_{bj} + g^1_j c_{sb} \}.
\]

We show that the absolute error of \( H_1 \) is \( c_{sb} \), and the relative error of \( H_1 \) is 1. Both bounds are tight. An average-case error bound for \( H_1 \) is also derived.
Let \( z^1 \equiv z^g \) be the solution value of the routing generated by H1, and \( z^{LP} \) be the solution value of the linear relaxation of R. The routing by an LP solution ships all of an item along a given path. For each item \( j \in N \), if \( c_{aj} \leq c_{aj} + a_j c_{ab} \), then the item is shipped direct with cost \( c_{aj} \). Otherwise, the item is shipped through the breakbulk with the proportional cost \( c_{aj} + a_j c_{ab} \). Therefore,

**Remark 3.1** \( z^{LP} = \sum_{j \in G_1} c_{aj} + \sum_{j \in G_1} (c_{aj} + a_j c_{ab}) \).

To establish the error bounds for H1, some preliminary results concerning the cost of various routes are needed.

**Lemma 3.1** If \( \sum_{j=1}^{n} c_{aj} \leq \sum_{j=1}^{n} c_{bj} + c_{ab} \), then \( z^* = \sum_{j=1}^{n} c_{aj} \). Otherwise, \( z^* \geq \sum_{j=1}^{n} c_{bj} + c_{ab} \).

**Proof.** There exists a set \( D^* \subseteq N \) such that an optimal routing ships each item \( j \in D^* \) directly to \( j \), and ships all other items through \( b \). If \( D^* = N \), then \( z^* = \sum_{j=1}^{n} c_{aj} \). Otherwise,

\[
\begin{align*}
z^* &= \sum_{j \in D^*} c_{aj} + \sum_{j \in D^*} c_{bj} + \left[ \sum_{j \in D^*} a_j \right] c_{ab} \\
&\geq \sum_{j=1}^{n} c_{aj} + \left[ \sum_{j \in D^*} a_j \right] c_{ab} \\
&\geq \sum_{j=1}^{n} c_{aj} + c_{ab}.
\end{align*}
\]

The first inequality follows because \( c_{aj} \geq c_{bj} \) for all \( j \). \( \square \)

Now, we perform a worst-case analysis for H1. We show that H1 has a relative error of 1, and the bound is tight. Theorem 3.4 shows that H1 gives near optimal solutions as \( n \to \infty \) if some problem parameters are bounded.
Theorem 3.2 $z^g - z^{LP} \leq c_{ab}$.

Proof. From Step 2 of $\mathcal{H}(g)$,

$$z^g \leq \sum_{j \in G} c_{aj} + \sum_{j \in G} c_{bj} + \left( \sum_{j \in \tilde{G}} a_j \right)c_{ab}.$$

Hence, from Remark 3.1,

$$z^g - z^{LP} \leq \sum_{j \in G} c_{aj} + \sum_{j \in G} c_{bj} + \left( \sum_{j \in \tilde{G}} a_j \right)c_{ab} - \left( \sum_{j \in \tilde{G}_1} c_{aj} + \sum_{j \in \tilde{G}_1} c_{bj} + \sum_{j \in \tilde{G}} a_j c_{ab} \right).$$

$$= \left( \sum_{j \in \tilde{G}} a_j \right)c_{ab} + \left( \sum_{j \in \tilde{G}} c_{aj} - \sum_{j \in \tilde{G}_1} c_{aj} \right)$$

$$+ \left( \sum_{j \in \tilde{G}} c_{bj} - \sum_{j \in \tilde{G}_1} c_{bj} \right).$$

(3.1)

Observe that, for parameters $e_1, e_2, \ldots, e_n$,

$$\sum_{j \in G} e_j - \sum_{j \in \tilde{G}_1} e_j = \sum_{j \in \tilde{G}} e_j - \sum_{j \in \tilde{G}_1 \setminus \tilde{G}} e_j = \sum_{j \in \tilde{G}} e_j - \sum_{j \in G} e_j. \quad (3.2)$$

Using (3.2), we rewrite some of the terms on the right hand side of (3.1). Therefore,

$$z^g - z^{LP} \leq \left( \sum_{j \in \tilde{G}} a_j \right)c_{ab}$$

$$+ \left( \sum_{j \in \tilde{G}} c_{aj} - \sum_{j \in \tilde{G}_1 \setminus \tilde{G}} c_{aj} \right)$$

$$+ \left( \sum_{j \in \tilde{G}} c_{bj} - \sum_{j \in \tilde{G}_1 \setminus \tilde{G}} c_{bj} \right)$$

$$= \left( \sum_{j \in \tilde{G}} a_j \right)c_{ab}$$

$$+ \sum_{j \in \tilde{G}} \left( -a_j c_{ab} + c_{aj} - c_{bj} \right)$$

$$+ \sum_{j \in \tilde{G}_1 \setminus \tilde{G}} (a_j c_{ab} - c_{aj} + c_{bj}).$$

From the definition of $G$, $g_j c_{ab} \geq c_{aj} - c_{bj}$ for $j \in G$, and $g_j c_{ab} < c_{aj} - c_{bj}$ for $j \in \tilde{G}$. Hence,

$$z^g - z^{LP} \leq \left( \sum_{j \in \tilde{G}} a_j \right)c_{ab}$$

$$+ \sum_{j \in \tilde{G}} \left( g_j a_j - a_j \right) + \sum_{j \in \tilde{G}_1 \setminus \tilde{G}} \left( a_j - g_j \right) c_{ab}. \quad (3.3)$$
Since \( g^j_j = a_j \) for all \( j \), we have
\[
z^1 - z^{LP} \leq \left( \sum_{j \in G_1} a_j \right) c_{sb} \leq c_{sb}. \quad \Box \tag{3.3}
\]

**Theorem 3.3** \( H_1 \) has an absolute error of \( c_{sb} \) and a relative error of 1. Moreover, the bounds are tight.

**Proof.** From Theorem 3.2, since \( z^* \geq z^{LP} \),
\[
z^1 - z^* \leq c_{sb}.
\]

Further by Lemma 3.1, if \( \sum_{j=1}^n c_{aj} \leq \sum_{j=1}^n c_{bj} + c_{sb} \), then both optimal and heuristic solutions ship all item \( j \) directly from \( s \) to \( j \) and the relative error is zero. Otherwise, \( z^* \geq \sum_{j=1}^n c_{bj} + c_{sb} \) and the relative error is
\[
\frac{z^1 - z^*}{z^*} \leq \frac{z^1 - z^*}{\sum_{j=1}^n c_{bj} + c_{sb}} \leq \frac{c_{sb}}{\sum_{j=1}^n c_{bj} + c_{sb}} \leq 1. \tag{3.4}
\]

To show that the bounds are tight, we construct an example with three items. The sizes of the items and the shipping costs are given in Table 3.1. For this example,

<table>
<thead>
<tr>
<th>( j )</th>
<th>( a_j )</th>
<th>( c_{aj} )</th>
<th>( c_{bj} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \frac{1}{2} )</td>
<td>( c_{sb} )</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>( \frac{1}{2} )</td>
<td>( c_{sb} )</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>( \varepsilon )</td>
<td>( 2\varepsilon c_{sb} )</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.1: Worst-case example for \( H_1 \).

\( G_1 = \emptyset \). Thus, either all items are shipped directly or all go through the breakbulk.
The solution value found by H1 is

\[
z^1 = \min\{c_{a1} + c_{a2} + c_{a3}, \ c_{b1} + c_{b2} + 2c_{sb}\}
\]
\[
= \min\{(2 + 2\varepsilon)c_{sb}, \ 2c_{sb}\}
\]
\[
= 2c_{sb}.
\]

The optimal routing ships items 1 and 2 through the breakbulk, and item 3 directly.

The cost is

\[
z^* = c_{b1} + c_{b2} + c_{sb} + c_{a3} = (1 + 2\varepsilon)c_{sb}.
\]

Consequently,

\[
z^1 - z^* = 2c_{sb} - (1 + 2\varepsilon)c_{sb}
\]
\[
= (1 - 2\varepsilon)c_{sb} \to c_{sb} \text{ as } \varepsilon \to 0.
\]

Also,

\[
\frac{z^1 - z^*}{z^*} = \frac{(1 - 2\varepsilon)c_{sb}}{(1 + 2\varepsilon)c_{sb}} \to 1 \text{ as } \varepsilon \to 0. \quad \square
\]

**Theorem 3.4** The relative error for H1 goes to 0 as \(\sum_{j=1}^n c_{bj}/c_{sb} \to \infty\).

**Proof.** From equation (3.4),

\[
\frac{z^1 - z^*}{z^*} \leq \frac{c_{sb}}{\sum_{j=1}^n c_{bj} + c_{sb}} = \frac{1}{\sum_{j=1}^n \frac{c_{bj}}{c_{sb}} + 1}.
\]

Hence, \((z^1 - z^*)/z^* \to 0\) as \(\sum_{j=1}^n c_{bj}/c_{sb} \to \infty\). \quad \square

Suppose that \(c_{bj}/c_{sb}\) is bounded away from zero for \(j \in J \subseteq N\) and that \(|J| \to \infty\) as \(n \to \infty\). Then, Theorem 3.4 implies that \((z^1 - z^*)/z^* \to 0\) as \(n \to \infty\).

**Theorem 3.5** H1 has the minimum worst-case relative error among all heuristics in \(\mathcal{H}(g)\).

22
Proof. Let $\varepsilon_1, \varepsilon_2, \ldots$ be an infinite sequence of positive real numbers such that $1/2 \geq \varepsilon_1, \varepsilon_i > \varepsilon_{i+1}$ for $i = 1, 2, \ldots,$ and $\varepsilon_i \to 0$ as $i \to \infty$. For a heuristic in $H(g)$, we construct a series of instances of $R$ with $n + 2$ items. Let $c_{kj} = 0$ for $j \in N$, $a_1 = a_2 = 1/2$, and $a_3 = a_4 = \ldots = a_{n+2} = \varepsilon$ where $\varepsilon = \varepsilon_1, \varepsilon_2, \ldots$. The proof is divided into four cases, and for each case we show that $(z^g - z^*)/z^* \to 1$ as $\varepsilon_i \to 0$.

Case 1: $g_l \geq 1$ for some $n \geq 1$ and some sequence $\{\varepsilon_i\}$.

Let $c_{s1} = g_l c_{sb}, c_{s2} = c_{sb}, c_{s3} = \varepsilon c_{sb},$ and $c_{sj} = 0$ for $j = 4, 5, \ldots, n + 2$. If $g_2 \geq 1$, then $\{1, 2\} \subseteq G$ and $\sum_{j \in G} c_{sj} \geq 2c_{sb}$. Otherwise, $\{1\} \subseteq G$ and $\sum_{j \in G} c_{sj} + [\sum_{j \in G} a_j] c_{sb} \geq 2c_{sb}$. Also, because $\sum_{j=1}^{n+2} c_{sj} = (1 + g_l + \varepsilon)c_{sb} > 2c_{sb}$ and $[\sum_{j=1}^{n+2} a_j] c_{sb} = 2c_{sb}$,

$$z^g = \min \left\{ \sum_{j=1}^{n+2} c_{sj}, \sum_{j=1}^{n+2} c_{sb} + [\sum_{j=1}^{n+2} a_j] c_{sb}, \sum_{j \in G} c_{sj} + \sum_{j \in G} c_{sb} + [\sum_{j \in G} a_j] c_{sb} \right\} = 2c_{sb}. $$

Since the optimal routing ships items 1 and 2 through the breakbulk and all other items direct,

$$z^* = (1 + \varepsilon)c_{sb}. $$

Therefore, for each $i \geq 1$,

$$\frac{z^g - z^*}{z^*} = \frac{2c_{sb} - (1 + \varepsilon_i)c_{sb}}{(1 + \varepsilon_i)c_{sb}} \to 1 \text{ as } \varepsilon_i \to 0.$$

Case 2: $g_2 \geq 1$ for some $n \geq 1$ and some sequence $\{\varepsilon_i\}$.

The proof is similar to Case 1.

Case 3: $g_l < 1, g_2 < 1$ for some $n \geq 1$ and some sequence $\{\varepsilon_i\}$, and $g_k \to 0$ as $\varepsilon_i \to 0$ for some $k \geq 3$.

Let $c_{s1} = c_{s2} = c_{sb}, c_{sk} = 2g_k c_{sb}$ and $c_{sj} = 0$ for $j = 3, 4, \ldots, n + 2$ and $j \neq k$. Thus, $G = \{1, 2, k\}$, and

$$z^g = \min \left\{ (2 + 2g_k)c_{sb}, 2c_{sb}, 2c_{sb} \right\} = 2c_{sb}. $$

23
The optimal routing ships items 1 and 2 through the breakbulk and all other items directly with the cost

\[ z^* = c_{ab} + 2 \sum_{j \in G} g_j c_{ab}. \]

Therefore,

\[ \frac{z^g - z^*}{z^*} = \frac{2c_{ab} - (c_{ab} + 2 \sum_{j \in G} g_j c_{ab})}{c_{ab} + 2 \sum_{j \in G} g_j c_{ab}} \rightarrow 1 \quad \text{as } \epsilon_i \rightarrow 0. \]

**Case 4:** \( g_1 < 1, g_2 < 1 \) for all \( n \geq 1 \) and some sequence \( \{\epsilon_i\} \), and \( g_j \geq K > 0 \) for \( j = 3, 4, \ldots, n+2 \) where \( K \) is a constant.

Let \( c_{s1} = c_{sb}, c_{s2} = 0, \) and \( c_{sj} = Kc_{sb} \) for \( j = 3, 4, \ldots, n+2 \). Thus, \( G = \{2, 3, \ldots, n+2\} \).

Let \( n \geq 1/K \). Then,

\[ z^g = \min \{c_{sb} + nKc_{sb}, 2c_{sb}, nKc_{sb} + c_{sb} \} = 2c_{sb}, \]

while the optimal routing ships item 2 direct and all other items through the breakbulk with the cost

\[ z^* = \left[ \frac{1}{2} + \epsilon \right] c_{sb} = c_{sb}. \]

Therefore,

\[ \frac{z^g - z^*}{z^*} = 1. \]

By Theorem 3.3, the relative error of H1 is 1. Consequently, H1 has the minimum worst-case relative error among all heuristics in \( \mathcal{H}(g) \). \( \Box \)

Although H1 has the minimum worst-case relative error in \( \mathcal{H}(g) \), Theorem 3.6 and 3.7 show that there are situations where \( z^1 \leq z^g \) and other situations where \( z^1 \geq z^g \).

**Lemma 3.2**

\[ z^1 - z^g = \left( \left[ \sum_{j \in G_i} a_j \right] - \left[ \sum_{j \in G} a_j \right] \right) c_{ab} + \sum_{j \in G_i \setminus G} (c_{sj} - c_{bj}) - \sum_{j \in G_i \setminus G_i} (c_{sj} - c_{bj}). \]
Proof. From the definitions of $G, G_1$, and $\mathcal{H}(g)$,

$$z^1 - z^g = \sum_{j \in G_1} c_{sj} + \sum_{j \in G_1} c_{bj} + \left( \sum_{j \in G} a_j \right) c_{sb} - \left( \sum_{j \in G} c_{sj} + \sum_{j \in G} c_{bj} + \left( \sum_{j \in G} a_j \right) c_{sb} \right)$$

$$= \left( \sum_{j \in G_1} a_j \right) c_{sb} + \left( \sum_{j \in G_1} c_{sj} - \sum_{j \in G} c_{sj} \right) + \left( \sum_{j \in G_1} c_{bj} - \sum_{j \in G} c_{bj} \right).$$

Using equation (3.2),

$$z^1 - z^g = \left( \sum_{j \in \bar{G}} a_j \right) c_{sb} + \left( \sum_{j \in \bar{G}} c_{sj} - \sum_{j \in \bar{G} \setminus G} c_{sj} \right)$$

$$+ \left( \sum_{j \in \bar{G} \setminus G_1} c_{bj} - \sum_{j \in \bar{G} \setminus G} c_{bj} \right)$$

$$= \left( \sum_{j \in G_1} a_j \right) c_{sb} + \sum_{j \in G_1 \setminus G} (c_{sj} - c_{bj}) - \sum_{j \in G \setminus G_1} (c_{sj} - c_{bj}). \; \Box$$

Theorem 3.6 Suppose $g_j \leq a_j$ for $j \in N$. If $\bar{G}_1 = \emptyset$, then $z^1 \leq z^g$.

Proof. When $g_j \leq a_j$ for all $j$, $G \subseteq G_1$. Then, by Lemma 3.2,

$$z^1 - z^g = \sum_{j \in G \setminus G_1} (c_{sj} - c_{bj}) + \left( \sum_{j \in G_1} a_j \right) c_{sb}.$$ 

For each $j \in G_1$, $c_{sj} - c_{bj} \leq a_j c_{sb}$. Thus,

$$z^1 - z^g \leq \left( \sum_{j \in G_1 \setminus G} a_j + \left( \sum_{j \in G_1} a_j \right) - \left( \sum_{j \in G} a_j \right) \right) c_{sb}.$$ 

If $\bar{G}_1 = \emptyset$, then $\sum_{j \in G_1 \setminus G} a_j = 0$ and $\sum_{j \in G_1 \setminus G} a_j = \sum_{j \in G} a_j$. Hence,

$$z^1 - z^g \leq \left( \sum_{j \in G} a_j - \left( \sum_{j \in G} a_j \right) \right) c_{sb} \leq 0. \; \Box$$

Theorem 3.7 Suppose $a_k \leq g_k \leq a_k \left[ \sum_{j \in G_1} a_j \right] / \sum_{j \in G_1} a_j$ for $k \in N$. If $\bar{G} = \emptyset$, then $z^1 \geq z^g$.

Proof. If $a_k \leq g_k \leq a_k \left[ \sum_{j \in G_1} a_j \right] / \sum_{j \in G_1} a_j$ for $k \in N$, then $G_1 \subseteq G$ and

$$c_{sk} - c_{bk} \leq g_k c_{sb} \leq \frac{a_k \left[ \sum_{j \in G_1} a_j \right]}{\sum_{j \in G_1} a_j} c_{sb} \quad \text{for } k \in G.$$ 

25
From Lemma 3.2 and $G_1 \subseteq G,$

$$z^1 - z^g = \left( \left[ \sum_{j \in G_1} a_j \right] - \left[ \sum_{j \in G} a_j \right] \right) c_{ab} - \sum_{j \in G \setminus G_1} \left( c_{aj} - c_{bj} \right)$$

$$\geq \left( \left[ \sum_{j \in G_1} a_j \right] - \left[ \sum_{j \in G} a_j \right] - \sum_{k \in G \setminus G_1} \frac{a_k \sum_{j \in G_1} a_j}{\sum_{j \in G_1} a_j} \right) c_{ab}.$$

If $\tilde{G} = \emptyset,$ then $\left[ \sum_{j \in \tilde{G}} a_j \right] = 0$ and $\sum_{j \in G \setminus G_1} a_j = \sum_{j \notin G_1} a_j.$ Consequently,

$$z^1 - z^g \geq \left( \sum_{j \in G_1} a_j \right) - \left( \sum_{j \notin G_1} a_j \right) c_{ab} = 0. \qed$$

We next derive an average-case error bound for $H_1.$ The error goes to zero asymptotically as $n$ goes to infinity provided that $c_{ab}$ is bounded. To perform the average-case analysis, we define $P(A)$ to be the probability of event $A,$ and $E(A)$ to be the expected value of event $A.$ Lemmas 3.3 and 3.4 can be found in Arnold and Meeden (1976).

**Lemma 3.3** Suppose $X_1, X_2, \ldots, X_n$ is a sequence of $n$ i.i.d. random variables on the interval $[0, 1).$ If $X_1$ does not have its distribution concentrated on a set of form $\{0, \frac{1}{v}, \ldots, \frac{v-1}{v}\}$ for some $v,$ then the fractional part of $\sum_{j=1}^n X_i$ converges to a random variable uniformly distributed over $[0, 1).$

**Lemma 3.4** If $X_1$ and $X_2$ are independent random variables on $[0, 1)$ and $X_1$ is uniformly distributed, then $X_1$ and the fractional part of $X_1 + X_2$ are i.i.d. uniform random variables on $[0, 1).$

For $R,$ let $X = \sum_{j \in G_1} A_j - \sum_{j \in \tilde{G}} A_j$ and $Y = \sum_{j \in G} A_j - \sum_{j \notin G} A_j.$ Recall that $G_1 = \{ j \in N \mid c_{sj} \leq c_{sj} + a_j c_{ab} \}$ and $G = \{ j \in N \mid c_{sj} \leq c_{sj} + g_j c_{ab} \}.$ Lemmas 3.5 and 3.6 show that both $X$ and $Y$ converge to $\text{Uniform}[0, 1).$
Lemma 3.5 $X$ converges to Uniform\([0,1]\).

Proof. Because $A_j$, $j \in \bar{G}_1$, are i.i.d. random variables on $[0,1)$, $\sum_{j \in \bar{G}_1} A_j - \lfloor \sum_{j \in \bar{G}_1} A_j \rfloor$ converges to Uniform\([0,1]\) by Lemma 3.3. If $\sum_{j \in \bar{G}_1} A_j - \lfloor \sum_{j \in \bar{G}_1} A_j \rfloor = 0$, then $X = 0$. Otherwise,

$$X = \lfloor \sum_{j \in \bar{G}_1} A_j \rfloor - \sum_{j \in \bar{G}_1} A_j = 1 - \left( \sum_{j \in \bar{G}_1} A_j - \lfloor \sum_{j \in \bar{G}_1} A_j \rfloor \right) = 1 - X.$$ 

Hence, $X$ converges to Uniform\([0,1]\). \qed

Lemma 3.6 $Y$ converges to Uniform\([0,1]\).

Proof. Since $g_j$ only depends on $a_j$ for $j \in N$, $A_j$ for $j \in \bar{G}$ are i.i.d. random variables on $[0,1)$. By the same argument as in the proof of Lemma 3.5, $Y$ converges to Uniform\([0,1]\). \qed

Suppose the distribution of $a_1, a_2, \ldots, a_n$ are i.i.d. from Uniform\((0,1)\). For $j = 1, 2, \ldots, n$, let $A_j$, $C_{aj}$ and $C_{bj}$ be random variables from the distributions of $a_j$, $c_{aj}$ and $c_{bj}$, respectively. The next result presents average-case error bounds for H1.

Given any function $g$, we show that $E(z^1) \leq E(z^2)$ when $n$ is sufficiently large.

Theorem 3.8 Given any $c_{ab}$, H1 has an average relative error of $c_{ab}/(2 \sum_{j=1}^{n} E(C_{bj}) + 2c_{ab})$. Further, the expected absolute error of H1 is bounded by $c_{ab}/2$.

Proof. From (3.3) and $z^* \geq z^{LP}$,

$$z^1 - z^* \leq \left( \sum_{j \in \bar{G}_1} a_j - \sum_{j \in \bar{G}_1} A_j \right) c_{ab}.$$ 

Therefore, by Lemmas 3.5,

$$E(z^1) - E(z^*) \leq E \left( \sum_{j \in \bar{G}_1} A_j - \sum_{j \in \bar{G}_1} A_j \right) c_{ab} = \frac{1}{2} c_{ab}.$$ 


Thus, the absolute error of $H1$ is bounded by $c_{ab}/2$. By (3.4),

$$\frac{z^1 - z^*}{z^*} \leq \frac{z^1 - z^*}{\sum_{j=1}^{n} c_{bj} + c_{ab}}.$$

Hence,

$$\frac{E(z^1) - E(z^*)}{E(z^*)} < \frac{c_{ab}/2}{E(\sum_{j=1}^{n} G_{bj}) + c_{ab}}. \quad \Box$$

**Lemma 3.7** Given any $c_{ab} \geq 0$, $E(z^1) - E(z^g) \leq (E(X) - E(Y))c_{ab}$.

**Proof.** By Lemma 3.2,

$$z^1 - z^g = \left( \left[ \sum_{j \in G^1} a_j \right] - \left[ \sum_{j \in G} a_j \right] \right) c_{ab} + \left( \sum_{j \in G^1 \setminus G} (c_{aj} - c_{bj}) \right) - \left( \sum_{j \in G \setminus G^1} (c_{aj} - c_{bj}) \right).$$

Since $c_{aj} - c_{bj} \leq a_j c_{ab}$ when $j \in G^1 \setminus G$, and $c_{aj} - c_{bj} \geq a_j c_{ab}$ when $j \in G \setminus G^1$,

$$z^1 - z^g \leq \left( \left[ \sum_{j \in G^1} a_j \right] - \left[ \sum_{j \in G} a_j \right] + \sum_{j \in G^1 \setminus G} a_j - \sum_{j \in G \setminus G^1} a_j \right) c_{ab}$$

$$= \left( \left[ \sum_{j \in G^1} a_j \right] - \left[ \sum_{j \in G} a_j \right] + \sum_{j \in G^1 \setminus G} a_j - \sum_{j \in G \setminus G^1} a_j \right) c_{ab}$$

where the first equality follows from (3.2). Therefore,

$$E(z^1) - E(z^g) \leq E\left( \left[ \sum_{j \in G^1} A_j \right] - \left[ \sum_{j \in G} A_j \right] \right) c_{ab} - E\left( \sum_{j \in G^1} A_j - \sum_{j \in G} A_j \right) c_{ab}$$

$$= (E(X) - E(Y))c_{ab}. \quad \Box$$

**Theorem 3.9** $E(z^1) \leq E(z^g)$ for $n$ sufficiently large.

**Proof.** By Lemma 3.7,

$$E(z^1) - E(z^g) \leq (E(X) - E(Y))c_{ab}.$$
Also by Lemma 3.5 and 3.6, both \( X \) and \( Y \) converge to \( \text{Uniform}[0,1] \). This implies that \( E(X) \) and \( E(Y) \) go to \( 1/2 \) as \( n \) goes to infinity. Consequently,

\[
E(z^1) - E(z^2) \leq 0 \quad \text{as } n \to \infty. \quad \square
\]

**Remark 3.2** For any constant \( \Delta \in \mathbb{R} \), if \( g_j = a_j + \Delta \) for all \( j \), then \( E(z^1) \leq E(z^2) \) for \( n \) sufficiently large.

**Proof.** Since \( \Delta \) is a constant, the result follows from Theorem 3.9. \( \square \)

Assuming the destinations \( j \in N \) are randomly located around \( b \) and \( c_{sb} = 1 \), an empirical study examining Remark 3.2 is presented in Appendix A. The purpose of the study is to determine how large \( n \) has to be for Remark 3.2 to hold. Suppose for each \( j \in N \), \( c_{sj} \) is the distance between \( s \) and \( j \), and \( c_{bj} \) is the distance between \( b \) and \( j \). The study shows that \( z^1 < z^2 \) when \( n \geq 3 \) with \( |\Delta| \geq .01 \). Recall that \( \mathcal{H}(g) \) always finds an optimal solution when \( n \leq 2 \). Therefore, in this case, Remark 3.2 is true for all \( n \).

Observe that H1 does not use all of the problem information. It uses only the item size when determining the route for an item. Utilizing more problem information in \( \mathcal{H}(g) \) may result in a better heuristic solution. As an example, we can use information about the sizes of other items and/or relative shipping costs (the difference between shipping direct and consolidating through the breakbulk). Consider the case when there is a large empty space in the last vehicle going through the breakbulk. If we can fill this empty space with items shipped direct, then the total cost can be reduced. Thus, we present a heuristic in the following that attempts to ship more items than H1 through the breakbulk to fill the empty space of the last vehicle going through the breakbulk after applying H1.
Procedure H2

1. Apply procedure H1 to R.

   If \( z^1 \neq \sum_{j \in G} c_{aj} + \sum_{j \in G} c_{bj} + [\sum_{j \in G} a_j]c_{ab} \), then stop.

2. Re-index the items of \( G \) so that, for \( j, j + 1 \in G \),

   \[
   \frac{c_{aj} - c_{bj}}{a_j} \geq \frac{c_{aj+1} - c_{bj+1}}{a_{j+1}}.
   \]

3. Find

   \[
   k = \min\{i \in G \mid a_i \leq [\sum_{j \in G} a_j] - \sum_{j \in G} a_j\}.
   \]

   If \( k \) does not exist, then ship items \( j \in \bar{G} \) through the breakbulk and the remaining items direct, and stop.

4. Set \( \bar{G} = \bar{G} \cup \{k\} \) and \( G = G \setminus \{k\} \). Go to Step 3.

The time complexity of H2 is \( O(n) \). Because H2 improves H1, \( z^1 \geq z^2 \). The improvement is accomplished by attempting to fill up the last trailer going through the breakbulk in H1. Since \( c_{aj} - c_{bj} \leq a_j c_{ab} \) for \( j \in G \), the improvement is no more than \( c_{ab} \). An empirical study is conducted to evaluate H1 and H2, and results are given in Appendix B. We find that H2 has an average absolute error of no more than .43, and an average relative error less than .033 in all cases studied. Since the difference between H1 and H2 is less than .5, H1 also performs well on average, especially when \( n \) is large.

3.4 Conclusion

We study the routing strategies for the single origin, single breakbulk distribution systems. Although this research is useful as an analysis of a warehouse freight
distribution problem, it also provides some insights for the more complicated general freight delivery systems with more than one breakbulk. Since the problem is \( \mathcal{NP} \)-hard, two polynomial time heuristic procedures, H1 and H2, are proposed.

When determining the routes for each item in the single origin, single breakbulk problem, H1 uses only the information pertinent to the item. In most on-line or dynamic environments, no other information is available. Consequently, only such heuristics are applicable. However, H2 uses all of the problem information. Procedure H1 has the minimum worst-case bound among all heuristics in \( \mathcal{H}(g) \). Although H2 has better average-case performance than H1, experimental evidence indicates that the difference is small.
CHAPTER 4

GENERAL SINGLE BREAKBULK PROBLEM

Consider a many-to-many freight distribution network shown in Figure 4.1. There are one breakbulk terminal $b$, a set of origin terminals $i \in S = \{1, 2, \ldots, m_1\}$, and a set of destination terminals $j \in T = \{1, 2, \ldots, n_1\}$. For each $i$ and $j$, item $(i, j)$ has $a_{ij}$ trailer-loads to be delivered from origin $i$ to destination $j$. Let $\bar{c}_{ij}$ be the directed arc from $i$ to $j$, $\bar{s}_i$ be the directed arc from $i$ to $b$, and $\bar{t}_j$ be the directed arc from $b$ to $j$, where $i \in S$ and $j \in T$. Associated with each arc $\bar{c}_{ij}$, $\bar{s}_i$ and $\bar{t}_j$ is a non-negative cost per trailer $c_{ij}$, $s_i$ and $t_j$ respectively. For each item $(i, j)$, if $a_{ij} \geq 1$, then $\lfloor a_{ij} \rfloor$ full trailers are dispatched, and only the fractional part of the item is considered for consolidation. Both optimal and heuristic routings ship the full trailer loads direct if $c_{ij} \leq s_i + t_j$; otherwise, they are shipped through the breakbulk without consolidation. Consequently, we assume that $0 \leq a_j < 1$.

We define the decision variables

\[
x_{ij} = \begin{cases} 
1 & \text{if item } (i, j) \text{ shipped from } i \text{ through } b \text{ to } j, \\
0 & \text{otherwise}, 
\end{cases} \quad i \in S, \ j \in T,
\]

\[
y^i_i = \text{number of trailers required from } i \text{ to } b, \quad i \in S,
\]

\[
y^j_j = \text{number of trailers required from } b \text{ to } j, \quad j \in T.
\]

A formulation of this problem is

32
Minimize \[ z(x, y) = \sum_{i \in S} \sum_{j \in T} c_{ij}(1 - x_{ij}) + \sum_{i \in S} s_i y_i^i + \sum_{j \in T} t_j y_j^j \]

subject to

\[ \sum_{j \in T} a_{ij} x_{ij} \leq y_i^i, \quad i \in S \]
\[ \sum_{i \in S} a_{ij} x_{ij} \leq y_j^j, \quad j \in T \]
\[ x_{ij} \in \{0, 1\}, \quad i \in S, \ j \in T \]
\[ y_i^i, y_j^j \in \mathbb{Z}_+, \quad i \in S, \ j \in T. \]

4.1 Notation and Definitions

For any two sets, \( X \) and \( Y \), denote

\[ (X, Y) = \{(i, j) \mid i \in X, \ j \in Y\}. \]
For set $Z \subseteq (X,Y)$, denote

$$
\tilde{Z} = \{(i,j) \in (X,Y) | (i,j) \notin Z\},
$$

$$
Z(i,\cdot) = \{j \in Y | (i,j) \in Z\} \quad \text{for } i \in X,
$$

$$
Z(\cdot,j) = \{i \in X | (i,j) \in Z\} \quad \text{for } j \in Y.
$$

When there is no ambiguity, we abbreviate $Z(\cdot,j)$ or $Z(i,\cdot)$ by $Z$. For example, if $(i,j) \in (X,Y)$, then

$$
\sum_{i \in Z} w_{ij} \equiv \sum_{i \in Z(i,j)} w_{ij},
$$

$$
\sum_{j \in Z} w_{ij} \equiv \sum_{j \in Z(i,j)} w_{ij}.
$$

For any item shipped through an arc, we define the shipping cost for the item on the arc to be the proportionally shared cost based on the item size divided by the total size of all items shipped on the arc. As an example, if item $(i,j) \in (S,T)$ is shipped through arc $\bar{s}_i$, then the shipping cost of item $(i,j)$ on arc $\bar{s}_i$ is

$$
\left[ \sum_{(i,j) \in A^*_i} a_{ij} \right] \times \frac{a_{ij}s_i}{\sum_{(i,j) \in A^*_i} a_{ij}}
$$

where $A^*_i$ is the set of all items shipped through $\bar{s}_i$. Let

$$
(x^*, y^*) = \text{an optimal solution for } R_s,
$$

$$
z^* = \text{the cost of } (x^*, y^*)
$$

$$
D^* = \{(i,j) \in (S,T) | x^*_{ij} = 0\}.
$$

Similarly, define $(x^h, y^h)$, $z^h$ and $D^h$ for the appropriate heuristic procedure. We also use the following notation:

$$
G = \{(i,j) \in (S,T) | c_{ij} \leq a_{ij}(s_i + t_j)\}
$$

$$
S^* = \{i \in S \mid x^*_{ij} = 0 \quad \text{for each } j \in T\}
$$

$$
T^* = \{j \in T \mid x^*_{ij} = 0 \quad \text{for each } i \in S\}
$$

$$
S^h = \{i \in S \mid \sum_{j \in G} c_{ij} \leq \left[ \sum_{j \in G} a_{ij} \right] s_i + \sum_{j \in G} a_{ij} t_j\}
$$

$$
T^h = \{j \in T \mid \sum_{i \in G} c_{ij} \leq \sum_{i \in G} a_{ij} s_i + \left[ \sum_{i \in G} a_{ij} \right] t_j\}.
$$
4.2 The Complexity of $R_s$

In Chapter 3, we show that the recognition version of $R$ is binary $NP$-complete. Since $R_s$ is a generalization of $R$, the recognition version of $R_s$ is also $NP$-complete. We establish the complexity of $R_s$ by using a reduction from the following unary $NP$-complete Problem.

3-Partition Problem (Garey and Johnson, 1979). Given integers $q_1, q_2, \ldots, q_{3m}$ and $d \in \mathbb{Z}_+$ where $d/4 < q_j < d/2$ and $\sum_{j=1}^{3m} q_j = md$. Can $q_1, q_2, \ldots, q_{3m}$ be partitioned into $m$ disjoint sets $Q_1, Q_2, \ldots, Q_m$ such that $\sum_{j \in Q_i} q_j = d$ for $1 \leq i \leq m$?

Theorem 4.1 The recognition version of $R_s$ is unary $NP$-complete.

Proof. Given the 3-Partition Problem, consider an instance of the recognition version of $R_s$ where $S = \{1, 2, \ldots, m+1\}$ and $T = \{1, 2, \ldots, 3m + 1\}$. Let

$$a_{ij} = \frac{q_j}{d+1}, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, 3m,$$

$$a_{i,3m+1} = \frac{1}{d+1}, \quad i = 1, 2, \ldots, m,$$

$$a_{m+1,j} = 1 - \frac{q_j}{d+1}, \quad j = 1, 2, \ldots, 3m,$$

$$c_{ij} = \frac{q_i}{d+1}, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, 3m,$$

$$c_{i,3m+1} = L, \quad i = 1, 2, \ldots, m,$$

$$c_{m+1,j} = L, \quad j = 1, 2, \ldots, 3m,$$

$$s_i = \frac{m^2 d}{d+1}, \quad i = 1, 2, \ldots, m,$$

$$t_j = \frac{m^2 d}{d+1}, \quad j = 1, 2, \ldots, 3m,$$

$$s_{m+1} = 0,$$

$$t_{3m+1} = 0,$$

$$K = \frac{4m^2 d}{d+1} + \frac{m^2 d}{d+1} - \frac{md}{d+1}$$

where $L = \frac{4m^2 d}{d+1} + \frac{m^2 d}{d+1}$. We show that there exists a yes instance to the 3-Partition Problem if and only if there is a solution to $R_s$ with cost no larger than $K$. 

35
(⇒) If there is a solution, \( \{Q_1, Q_2, \ldots, Q_m\} \), to the 3-Partition problem such that \( \sum_{j \in Q_1} q_j = d \) for \( 1 \leq i \leq m \), then for each \( i = 1, 2, \ldots, m \), let

\[
x_{ij} = \begin{cases} 
1 & \text{for } j \in Q_i \\
0 & \text{for } j \in \bar{Q}_i.
\end{cases}
\]

Also, let

\[
x_{i,3m+1} = 1 \quad \text{for } i = 1, 2, \ldots, m
\]

\[
x_{m+1,j} = 1 \quad \text{for } j = 1, 2, \ldots, 3m.
\]

Thus, for each \( i = 1, 2, \ldots, m \),

\[
\sum_{j \in T} a_{ij} x_{ij} = \sum_{j \in Q_i} a_{ij} + a_{i,3m+1} = \sum_{j \in Q_i} q_j \frac{d}{d+1} + \frac{1}{d+1} = 1,
\]

and for each \( j = 1, 2, \ldots, 3m \),

\[
\sum_{i \in S} a_{ij} x_{ij} = \frac{q_j}{d+1} + \left(1 - \frac{q_j}{d+1}\right) = 1.
\]

Because \( s_{m+1} = 0 \), the cost of shipping the items which go through the breakbulk is

\[
\sum_{i \in S} s_i y_i^s + \sum_{j \in T} t_j y_j^f = \sum_{i=1}^{m} s_i + \sum_{i=1}^{3m} t_j = \frac{m^3d}{d+1} + \frac{3m^3d}{d+1} = \frac{4m^3d}{d+1}.
\]

(4.1)

Also, because \( \sum_{i=1}^{m} \sum_{j \in Q_1} a_{ij} = \frac{md}{d+1} \) and \( \sum_{i=1}^{m} \sum_{j=1}^{3m} a_{ij} = \frac{m^2d}{d+1} \), the cost of shipping the items which go directly from origins to destinations is

\[
\sum_{i \in S} \sum_{j \in T} c_{ij} (1 - x_{ij}) = \sum_{i=1}^{m} \sum_{j \in Q_i} c_{ij}
\]

36
\[ = \sum_{i=1}^{m} \sum_{j \in \Omega_i} a_{ij} \]
\[ = \frac{m^2d}{d+1} - \frac{md}{d+1}. \quad (4.2) \]

Consequently, from equations (4.1) and (4.2), the cost of the solution equals \( K \) and we have a \textit{yes} instance to the recognition version of \( R_s \).

(\( \Longleftrightarrow \)) Observe that \( c_{i,3m+1} > K \) for \( i = 1, 2, \ldots, m \) and \( c_{m+1,j} > K \) for \( j = 1, 2, \ldots, 3m \).

If there is a solution \((x, y)\) to \( R_s \) with cost no larger than \( K \), then \( x_{i,3m+1} = 1 \) for \( i = 1, 2, \ldots, m \) and \( x_{m+1,j} = 1 \) for \( j = 1, 2, \ldots, 3m \). The cost of shipping these items through \( b \) is
\[ \sum_{i=1}^{m} s_i + \sum_{i=1}^{3m} t_j = \frac{m^2d}{d+1} + \frac{3m^3d}{d+1} = \frac{4m^3d}{d+1}. \]

The remaining available trailer space is \( \frac{d}{d+1} \) for arc \( \bar{s}_i, i = 1, 2, \ldots, m \), and \( \frac{d}{d+1} \) for arc \( \bar{t}_j, j = 1, 2, \ldots, 3m \).

For each \( i = 1, 2, \ldots, m \), since
\[ s_i > \frac{m^2d}{d+1} - \frac{md}{d+1}, \]
the cost is larger than \( K \) if more than one trailer is used on \( \bar{s}_i \). By a similar argument, for any \( j \in \{1, 2, \ldots, 3m\} \), the cost is larger than \( K \) if more than one trailer is used on \( \bar{t}_j \). Observe that the cost is \( \frac{m^2d}{d+1} \) to ship direct all items \((i, j)\) for \( i = 1, 2, \ldots, m \) and \( j = 1, 2, \ldots, 3m \). Consequently, if there is a solution with a cost no larger than \( K \), then we ship items with total size of at least \( \frac{md}{d+1} \) through \( b \). Since \( \frac{md}{d+1} \) is the total trailer space available on \( \bar{t}_j \) for \( j = 1, 2, \ldots, 3m \), we must fill all of the remaining space available on each \( \bar{t}_j \). Also, because \( \frac{d}{d+1} \) is the remaining trailer space available on \( \bar{t}_j \) for \( j = 1, 2, \ldots, 3m \), one item must go to each destination. Further, because the total trailer space available on \( \bar{s}_i \) for \( i = 1, 2, \ldots, m \) is \( \frac{md}{d+1} \), we must also fill the available trailer space, \( \frac{d}{d+1} \), on each \( \bar{s}_i \) for \( i = 1, 2, \ldots, m \).
Let

\[ Q_i = \{ j \in \{1, 2, \ldots, 3m \} \mid x_{ij} = 1 \} \quad \text{for } i = 1, 2, \ldots, m. \]

Since \( \sum_{j \in Q_i} \frac{q_j}{d+1} = \frac{d}{d+1} \) for \( i = 1, 2, \ldots, m \), a solution to \( R_s \) implies that

\[ \sum_{j \in Q_i} q_j = d. \]

Therefore, \( \{Q_1, Q_2, \ldots, Q_m\} \) is a solution to the 3-Partition Problem. \( \square \)

### 4.3 A Heuristic Procedure

We develop a heuristic procedure \( HS \) for \( R_s \), and show that \( HS \) has a worst-case relative error of 1.

**Procedure HS**

1. Ship direct all items \( (i, j) \in G = \{(i, j) \in (S, T) \mid c_{ij} \leq a_{ij}(s_i + t_j)\} \).

2. For each \( i \in S \), ship items \( (i, j), j \in T \), direct if

\[ \sum_{j \in \mathcal{G}} c_{ij} \leq \left\lfloor \sum_{j \in \mathcal{G}} a_{ij} \right\rfloor s_i + \sum_{j \in \mathcal{G}} a_{ij} t_j. \]

For each \( j \in T \), ship items \( (i, j), i \in S \), direct if

\[ \sum_{i \in \mathcal{G}} c_{ij} \leq \sum_{i \in \mathcal{G}} a_{ij} s_i + \left\lceil \sum_{i \in \mathcal{G}} a_{ij} \right\rceil t_j. \]

3. Ship all other items through \( b \).

4. Calculate the cost \( z^H \). Stop.

For each item \( (i, j) \in G \), the direct shipping cost is no larger than the proportional cost for shipping the item through \( b \). Unless there are trailers dispatched with space available on the route that goes through the breakbulk, shipping the item \( (i, j) \in G \)
direct is cheaper. This is the motivation for Step 1 in HS. In Step 2, the first condition implies that, for each \( i \in S \), shipping all items \((i, j)\) for \( j \in G \) direct is cheaper than shipping them all through \( b \). Therefore, if the condition is satisfied, then HS ships all items \((i, j)\) from terminal \( i \) for \( j \in G \) directly. An equivalent condition is also checked for each destination terminal \( j \in T \).

Let \( n = m_1 \times n_1 \). Since there are at most \( n \) items to be delivered, each step in HS takes \( O(n) \) time to perform. Consequently, the time complexity of HS is \( O(n) \).

Next, we develop error bounds on the worst case performance of HS. Lemma 4.1 provides an upper bound on the heuristic solution.

**Lemma 4.1** \( z^H \leq \sum_{(i,j) \in G} c_{ij} + \sum_{(i,j) \in G} a_{ij}(s_i + t_j) + \sum_{i \in S} s_i + \sum_{j \in T} t_j \).

**Proof.** By definition of \( R_* \),

\[
z^H = \sum_{(i,j) \in G} c_{ij} + \sum_{(i,j) \in D^h} c_{ij} + \sum_{j \in S^h} \left[ \sum_{i \in B^h} a_{ij} \right] s_i + \sum_{i \in T^h} \left[ \sum_{j \in D^h} a_{ij} \right] t_j.
\]

(4.3)

For the second term on the right hand side of (4.3), observe that

\[
\sum_{(i,j) \in D^h \cap G} c_{ij} = \sum_{i \in S^h} \sum_{j \in D^h} c_{ij} + \sum_{j \in T^h} \sum_{i \in G} c_{ij} - \sum_{(i,j) \in (S^h,T^h) \cap G} c_{ij}.
\]

(4.4)

Because \( \sum_{j \in G} c_{ij} \leq \left[ \sum_{j \in G} a_{ij} \right] s_i + \sum_{j \in G} a_{ij} t_j \) for \( i \in S^h \),

\[
\sum_{i \in S^h} \sum_{j \in D^h} c_{ij} \leq \sum_{i \in S^h} \left( \left[ \sum_{j \in G} a_{ij} \right] s_i + \sum_{j \in G} a_{ij} t_j \right) \leq \sum_{i \in S^h} \left( s_i + \sum_{j \in G} a_{ij}(s_i + t_j) \right).
\]

(4.5)

Similarly,

\[
\sum_{j \in T^h} \sum_{i \in G} c_{ij} \leq \sum_{j \in T^h} \left( \sum_{i \in G} a_{ij} s_i + \left[ \sum_{i \in G} a_{ij} \right] t_j \right) \leq \sum_{j \in T^h} \left( t_j + \sum_{i \in G} a_{ij}(s_i + t_j) \right).
\]

(4.6)

Also, because \( c_{ij} > a_{ij}(s_i + t_j) \) for \((i, j) \in G\),

\[
\sum_{(i,j) \in (S^h,T^h) \cap G} c_{ij} > \sum_{(i,j) \in (S^h,T^h) \cap G} a_{ij}(s_i + t_j).
\]

(4.7)
Thus, by (4.4) - (4.7),
\[
\sum_{(i,j) \in \bar{D}^h \cap \bar{G}} c_{ij} \leq \sum_{i \in \bar{S}^h} \left( s_i + \sum_{j \in \bar{G}} a_{ij}(s_i + t_j) \right) + \sum_{j \in \bar{T}^h} \left( t_j + \sum_{i \in \bar{G}} a_{ij}(s_i + t_j) \right) - \sum_{(i,j) \in (\bar{S}^h, \bar{T}^h) \cap \bar{G}} a_{ij}(s_i + t_j)
\]
\[= \sum_{i \in \bar{S}^h} s_i + \sum_{j \in \bar{T}^h} t_j + \sum_{(i,j) \in \bar{D}^h \cap \bar{G}} a_{ij}(s_i + t_j) \tag{4.8}\]
where the last term in (4.8) follows from (4.4) with \(c_{ij}\) replaced by \(a_{ij}(s_i + t_j)\).

For the last two terms on the right hand side of (4.3), because \(\sum_{(i,j) \in \bar{D}^h} a_{ij} = \sum_{i \in \bar{S}^h} \sum_{j \in \bar{T}^h} a_{ij}\),
\[
\sum_{i \in \bar{S}^h} \left[ \sum_{j \in \bar{D}^h} a_{ij} \right] s_i + \sum_{j \in \bar{T}^h} \left[ \sum_{i \in \bar{D}^h} a_{ij} \right] t_j
\]
\[\leq \sum_{i \in \bar{S}^h} \left( 1 + \sum_{j \in \bar{D}^h} a_{ij} \right) s_i + \sum_{j \in \bar{T}^h} \left( 1 + \sum_{i \in \bar{D}^h} a_{ij} \right) t_j
\]
\[\leq \sum_{i \in \bar{S}^h} s_i + \sum_{j \in \bar{T}^h} t_j + \sum_{(i,j) \in \bar{D}^h} a_{ij}(s_i + t_j). \tag{4.9}\]

Hence, from equations (4.3), (4.8) and (4.9), because \(\bar{D}^h \subseteq \bar{G}\),
\[
z^h \leq \sum_{(i,j) \notin \bar{G}} c_{ij} + \sum_{(i,j) \notin \bar{G}} a_{ij}(s_i + t_j) + \sum_{i \in \bar{S}} s_i + \sum_{j \in \bar{T}} t_j. \quad \square
\]

Before we develop the error bounds for HS, an example is presented to illustrate the worst-case performance.

**Example 4.1** Consider an instance of \(R_s\) with two origins and two destinations.

There are four items \((1,1), (1,2), (2,1)\) and \((2,2)\) to be shipped. Let
\[
a_{11} = 2\varepsilon \\
a_{12} = 1 - \varepsilon \\
a_{21} = 1 - \varepsilon \\
a_{22} = 2\varepsilon \\
c_{11} = 2\varepsilon(s_1 + t_1) + \varepsilon \\
c_{12} = 2(s_1 + t_2) \\
c_{21} = 2(s_2 + t_1) \\
c_{22} = 2\varepsilon(s_2 + t_2) + \varepsilon
\]
where $0 < \varepsilon < \frac{s_1 + s_2 + t_1 + t_2}{2(s_1 + s_2 + t_1 + t_2 + 1)}$. The solution generated by HS ships all items through $b$ with cost

$$z^h = 2(s_1 + s_2 + t_1 + t_2).$$

In the optimal solution, $a_{11}, a_{22}$ are shipped directly, and $a_{12}, a_{21}$ are shipped through $b$ with cost

$$z^* = 2\varepsilon(s_1 + t_1) + \varepsilon + 2\varepsilon(s_2 + t_2) + \varepsilon + (s_1 + s_2 + t_1 + t_2)$$

$$= (1 + 2\varepsilon)(s_1 + s_2 + t_1 + t_2) + 2\varepsilon.$$

Thus, the absolute error is

$$z^h - z^* = 2(s_1 + s_2 + t_1 + t_2) - (1 + 2\varepsilon)(s_1 + s_2 + t_1 + t_2) - 2\varepsilon$$

$$= (1 - 2\varepsilon)(s_1 + s_2 + t_1 + t_2) - 2\varepsilon.$$

Consequently, as $\varepsilon \to 0$, $z^h - z^* \to s_1 + s_2 + t_1 + t_2$. Note that, the error is the cost of one trailer on each arc that goes to and from the breakbulk. The relative error is

$$\frac{z^h - z^*}{z^*} = \frac{(1 - 2\varepsilon)(s_1 + s_2 + t_1 + t_2) - 2\varepsilon}{(1 + 2\varepsilon)(s_1 + s_2 + t_1 + t_2) + 2\varepsilon}.$$

As $\varepsilon \to 0$, $(z^h - z^*)/z^* \to 1$.

Theorem 4.2 provides an absolute error, and Theorem 4.3 provides a relative error bound for HS. Both bounds are tight.

**Theorem 4.2** The absolute error of the solution generated by HS is no larger than $\sum_{i \in S} s_i + \sum_{j \in T} t_j$. Further, the bound is tight.

**Proof.** The total cost of an optimal routing for $R_s$ is

$$z^* = \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S} \sum_{j \in D^*} a_{ij} s_i + \sum_{j \in T} \sum_{i \in D^*} a_{ij} t_j$$

41
\[
\sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S} \sum_{j \in D^*} a_{ij} s_i + \sum_{j \in T} \sum_{i \in D^*} a_{ij} t_j \\
= \sum_{(i,j) \in D^*} c_{ij} + \sum_{(i,j) \in D^*} a_{ij} (s_i + t_j) \\
\geq \sum_{(i,j) \in D^* \cap G} c_{ij} + \sum_{(i,j) \in D^* \cap G} a_{ij} (s_i + t_j) + \sum_{(i,j) \in D^* \cap G} c_{ij} + \sum_{(i,j) \in D^* \cap G} a_{ij} (s_i + t_j) \\
= \sum_{(i,j) \in G} c_{ij} + \sum_{(i,j) \in G} a_{ij} (s_i + t_j).
\]

Using Lemma 4.1, the absolute error of the solution generated by HS is

\[
z^h - z^* \leq \sum_{i \in S} s_i + \sum_{j \in T} t_j.
\]

From Example 4.1, the bound is tight. □

Given an instance of \( R_s \), suppose there exist items \( j \in D^* \cap G \) that are shipped direct by both optimal and HS routings. We convert the instance to a problem called \( \hat{R}_s \) by removing all items in \( D^* \cap G \) from \( R_s \). The following lemma shows the relationship between the relative error bounds of HS for \( \hat{R}_s \) and \( R_s \).

**Lemma 4.2** If \( \rho \) is a relative error bound of the solution generated by HS for \( \hat{R}_s \), then \( \rho \) is also a relative error bound of HS for \( R_s \).

**Proof.** For items \( (i,j) \in D^* \cap G \), both optimal and heuristic solutions ship these items directly from origin to destination with the identical cost

\[
K = \sum_{(i,j) \in D^* \cap G} c_{ij}.
\]

Let \( \hat{z}^* \) be the cost of an optimal solution, and \( \hat{z}^h \) be the cost of the heuristic solution generated by HS to \( \hat{R}_s \). Since the items \( (i,j) \in D^* \cap G \) are shipped direct and do not affect the consolidation for other items in \( R_s \),

\[
\frac{z^h - z^*}{z^*} = \frac{(\hat{z}^h + K) - (\hat{z}^* + K)}{\hat{z}^* + K} \\
\leq \frac{\hat{z}^h - \hat{z}^*}{\hat{z}^*} \leq \rho. \quad \square
\]

42
From Lemma 4.2, we assume that \( D^* \cap G = \emptyset \) in the analysis of the worst case error. Also, observe that if \( D^* \cap G = \emptyset \), then \( D^* \cap \bar{G} = D^* \) and \( \bar{D}^* \cap G = G \).

Next, we develop a lower bound for the optimal solution and an upper bound for the heuristic solution generated by HS. Then, we combine these results to obtain a bound on the relative error for HS.

\[ 2z^* \geq \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S^* \cap \bar{S}^H} \sum_{j \in \bar{G}} c_{ij} + \sum_{j \in T^* \cap T^H} \sum_{i \in \bar{G}} c_{ij} + \sum_{i \in \bar{S}^* \cup \bar{S}^H} \left( s_i + \sum_{j \in \bar{G}} a_{ij} t_j \right) + \sum_{j \in \bar{T}^* \cup \bar{T}^H} \left( t_j + \sum_{i \in \bar{G}} a_{ij} s_i \right). \]

**Proof.** Note that

\[ \sum_{(i,j) \in D^*} a_{ij} = \sum_{i \in S^*} \sum_{j \in D^*} a_{ij} = \sum_{j \in T^*} \sum_{i \in \bar{D}^*} a_{ij}. \]

Then, the total shipping cost for an optimal solution is

\[ z^* = \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S^*} [ \sum_{j \in \bar{D}^*} a_{ij} ] s_i + \sum_{j \in T^*} [ \sum_{i \in \bar{D}^*} a_{ij} ] t_j \]
\[ \geq \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S^*} s_i + \sum_{j \in T^*} \sum_{i \in \bar{D}^*} a_{ij} t_j \]
\[ = \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S^*} s_i + \sum_{(i,j) \in D^*} a_{ij} t_j \]
\[ = \sum_{(i,j) \in D^* \cap \bar{G}} c_{ij} + \sum_{i \in S^*} s_i + \sum_{(i,j) \in \bar{D}^* \cap \bar{G}} a_{ij} t_j + \sum_{(i,j) \in D^* \cap G} a_{ij} t_j + \sum_{i \in S^*} \sum_{j \in \bar{D}^* \cap G} a_{ij} t_j. \]

Since \( S^* \subseteq D^*(\cdot, j) \) for each \( j \in N \), observe that

\[ \sum_{(i,j) \in D^* \cap \bar{G}} c_{ij} = \sum_{i \in S^* \cap \bar{S}^H} \sum_{j \in \bar{G}} c_{ij} + \sum_{i \in \bar{S}^* \cap \bar{S}^H} \sum_{j \in \bar{G}} c_{ij} + \sum_{i \in S^* \cap \bar{D}^* \cap \bar{G}} \sum_{j \in \bar{G}} c_{ij} \]
\[ > \sum_{i \in S^* \cap \bar{S}^H} \sum_{j \in \bar{G}} c_{ij} + \sum_{i \in S^* \cap \bar{S}^H} \left[ \sum_{j \in \bar{G}} a_{ij} s_i + \sum_{j \in \bar{G}} a_{ij} t_j \right]. \]
\[ z^* > \sum_{i \in S^* \cap S^h} \sum_{j \in \overline{G}} c_{ij} + \sum_{i \in S^* \cap S^h} \left( s_i + \sum_{j \in \overline{G}} a_{ij}t_j \right) \]
\[ + \sum_{i \in S^* \cup S^h} \sum_{j \in \overline{G}} a_{ij}t_j + \sum_{(i,j) \in \overline{G}} a_{ij}t_j + \sum_{i \in S^* \cap S^h} \sum_{j \in \overline{G}} a_{ij}t_j \]
\[ = \sum_{i \in S^* \cap S^h} \sum_{j \in \overline{G}} c_{ij} + \sum_{i \in S^* \cap S^h} \left( s_i + \sum_{j \in \overline{G}} a_{ij}t_j \right) \]
\[ + \sum_{i \in S^* \cup S^h} \left( s_i + \sum_{j \in \overline{G}} a_{ij}t_j \right) + \sum_{(i,j) \in \overline{G}} a_{ij}t_j \]
\[ = \sum_{i \in S^* \cap S^h} \sum_{j \in \overline{G}} c_{ij} + \sum_{i \in S^* \cup S^h} \left( s_i + \sum_{j \in \overline{G}} a_{ij}t_j \right) + \sum_{(i,j) \in \overline{G}} a_{ij}t_j. \]

By a similar argument,

\[ z^* > \sum_{j \in T^* \cap T^h} \sum_{i \in \overline{G}} c_{ij} + \sum_{j \in T^* \cup T^h} \left( t_j + \sum_{i \in \overline{G}} a_{ij}s_i \right) + \sum_{(i,j) \in \overline{G}} a_{ij}s_i. \]

Consequently,

\[ 2z^* > \sum_{i \in S^* \cap S^h} \sum_{j \in \overline{G}} c_{ij} + \sum_{j \in T^* \cap T^h} \sum_{i \in \overline{G}} c_{ij} + \sum_{(i,j) \in \overline{G}} a_{ij}(s_i + t_j) \]
\[ + \sum_{i \in S^* \cup S^h} \left( s_i + \sum_{j \in \overline{G}} a_{ij}t_j \right) + \sum_{j \in T^* \cup T^h} \left( t_j + \sum_{i \in \overline{G}} a_{ij}s_i \right) \]
\[ \geq \sum_{(i,j) \in \overline{G}} c_{ij} + \sum_{i \in S^* \cap S^h} \sum_{j \in \overline{G}} c_{ij} + \sum_{j \in T^* \cap T^h} \sum_{i \in \overline{G}} c_{ij} \]
\[ + \sum_{i \in S^* \cup S^h} \left( s_i + \sum_{j \in \overline{G}} a_{ij}t_j \right) + \sum_{j \in T^* \cup T^h} \left( t_j + \sum_{i \in \overline{G}} a_{ij}s_i \right). \]
Lemma 4.4

\[ z^h \leq \sum_{(i,j) \in D^h \cap G} c_{ij} + \sum_{i \in S^h} \sum_{j \in G} c_{ij} + \sum_{j \in T^h} \sum_{i \in G} c_{ij} + \sum_{j \in T^h} \sum_{i \in G} c_{ij} \]
\[ + \sum_{i \in S^h} s_i + \sum_{j \in G} t_j + \sum_{i \in S^h} \sum_{j \in G} a_{ij}t_j + \sum_{j \in T^h} \sum_{i \in G} a_{ij}s_i. \]

**Proof.** By expanding equation (4.4), we obtain

\[ \sum_{(i,j) \in D^h \cap G} c_{ij} = \sum_{i \in S^h} \sum_{j \in G} c_{ij} + \sum_{j \in T^h} \sum_{i \in G} c_{ij} + \sum_{i \in S^h} \sum_{j \in G} c_{ij} - \sum_{(i,j) \in (S^h, T^h) \cap G} c_{ij}. \quad (4.11) \]

By the same argument as in (4.5) and (4.6),

\[ \sum_{i \in S^h} \sum_{j \in G} c_{ij} \leq \sum_{i \in S^h} \left( s_i + \sum_{j \in G} a_{ij}(s_i + t_j) \right), \quad (4.12) \]

\[ \sum_{j \in T^h} \sum_{i \in G} c_{ij} \leq \sum_{j \in T^h} \left( t_j + \sum_{i \in G} a_{ij}(s_i + t_j) \right). \quad (4.13) \]

Using (4.11), and then (4.12), (4.13) and (4.7),

\[ \sum_{(i,j) \in D^h \cap G} c_{ij} \leq \sum_{i \in S^h} \sum_{j \in G} c_{ij} + \sum_{j \in T^h} \sum_{i \in G} c_{ij} + \sum_{i \in S^h} \left( s_i + \sum_{j \in G} a_{ij}(s_i + t_j) \right) \]
\[ + \sum_{j \in T^h} \left( t_j + \sum_{i \in G} a_{ij}(s_i + t_j) \right) - \sum_{(i,j) \in (S^h, T^h) \cap G} a_{ij}(s_i + t_j) \]
\[ = \sum_{i \in S^h} \sum_{j \in G} c_{ij} + \sum_{j \in T^h} \sum_{i \in G} c_{ij} + \sum_{i \in S^h} s_i + \sum_{j \in T^h} t_j \]
\[ + \sum_{(i,j) \in D^h \cap G} a_{ij}(s_i + t_j) - \sum_{i \in S^h} \sum_{j \in G} a_{ij}(s_i + t_j) \]
\[ - \sum_{j \in T^h} \sum_{i \in G} a_{ij}(s_i + t_j). \quad (4.14) \]

Equality in (4.14) is obtained from (4.11) by replacing \( c_{ij} \) with \( a_{ij}(s_i + t_j) \) and rearranging the terms. Hence, from equation (4.3), then from (4.14) and (4.9), and then because \( \bar{D}^h \subset \bar{G} \),

\[ z^h = \sum_{(i,j) \in G} c_{ij} + \sum_{(i,j) \in D^h \cap G} c_{ij} + \sum_{i \in S^h} \sum_{j \in D^h} a_{ij} s_i + \sum_{j \in T^h} \sum_{i \in D^h} a_{ij} t_j \]

45
\[ \begin{align*}
\leq & \sum_{(i,j) \in G} c_{ij} + \sum_{i \in S^h} \sum_{j \in G} c_{ij} + \sum_{j \in T^h} \sum_{i \in E^h} c_{ij} + \sum_{i \in S^h} s_i + \sum_{j \in T^h} t_j \\
& + \sum_{(i,j) \in D^h} a_{ij}(s_i + t_j) - \sum_{i \in S^h} \sum_{j \in G} a_{ij}(s_i + t_j) \\
& - \sum_{j \in T^h} \sum_{i \in E^h} a_{ij}(s_i + t_j) + \sum_{i \in S^h} s_i + \sum_{j \in T^h} t_j + \sum_{(i,j) \in D^h} a_{ij}(s_i + t_j) \\
= & \sum_{(i,j) \in G} c_{ij} + \sum_{i \in S^h} \sum_{j \in G} c_{ij} + \sum_{j \in T^h} \sum_{i \in E^h} c_{ij} + \sum_{i \in S^h} s_i + \sum_{j \in T^h} t_j \\
& + \sum_{(i,j) \in G} a_{ij} t_j - \sum_{i \in S^h} \sum_{j \in G} a_{ij} t_j + \sum_{(i,j) \in G} a_{ij} s_i - \sum_{j \in T^h} \sum_{i \in E^h} a_{ij} s_i \\
= & \sum_{(i,j) \in G} c_{ij} + \sum_{i \in S^h} \sum_{j \in G} c_{ij} + \sum_{j \in T^h} \sum_{i \in E^h} c_{ij} + \sum_{i \in S^h} s_i + \sum_{j \in T^h} t_j \\
& + \sum_{i \in S^h} \sum_{j \in G} a_{ij} t_j + \sum_{j \in T^h} \sum_{i \in E^h} a_{ij} s_i. \quad \Box
\end{align*} \]

**Theorem 4.3** The relative error of the solution generated by HS is 1. Further, the bound is tight.

**Proof.** From Lemma 4.3 and 4.4, we obtain \(2z^* > z^h\). This implies that

\[
\frac{z^h - z^*}{z^*} \leq 1.
\]

From Example 4.1, the bound is tight. \(\Box\)

### 4.4 Conclusion

We extend the single origin freight delivery problem so that there are multiple origins and destinations. We show that this problem is unary \(\mathcal{NP}\)-hard. An \(O(n)\) time heuristic procedure, HS, extended from H1 is proposed. The absolute error for HS is equal to the total cost of dispatching one trailer on each arc going to and from the breakbulk, and the relative error for HS is 1.
CHAPTER 5

CLIQUE BREAKBULK PROBLEM

Consider a many-to-many freight delivery network as shown in Figure 5.1 that has a set of breakbulk terminals \( k \in B = \{1, 2, \ldots, b\} \). For each \( k \in B \), there are a set of origin terminals \( i \in S_k = \{m_{k-1} + 1, m_{k-1} + 2, \ldots, m_k\} \), and a set of destination terminals \( j \in T_k = \{n_{k-1} + 1, n_{k-1} + 2, \ldots, n_k\} \) where \( m_0 = n_0 = 0 \). Let \( S = \bigcup_{k \in B} S_k \) and \( T = \bigcup_{l \in B} T_l \). Thus, \( |S| = m_b \) and \( |T| = n_b \). We say that \( k \in B \) is the primary breakbulk of \( i \in S_k \), \( l \in B \) is the primary breakbulk of \( j \in T_l \), and \( k, l \) are the primary breakbulks of item \((i, j)\). For each \( i \) and \( j \), item \((i, j)\) has \( a_{ij} \) trailer-loads to be delivered from origin \( i \) to destination \( j \) through a set of directed arcs. We assume that the breakbulks form a clique where an arc exists between all breakbulk pairs, and items can only be shipped direct or through their primary breakbulks and the arc between the breakbulks. That is, each item \((i, j)\) can only be shipped from origin \( i \) either directly to destination \( j \) or through breakbulks \( k, l \), and then to \( j \) where \((i, j) \in (S_k, T_l)\). For each \((i, j) \in (S_k, T_l)\), \( k, l \in B \), let

\[
\begin{align*}
\bar{c}_{ij} &= \text{the directed arc from origin } i \text{ to destination } j, \\
\bar{s}_i &= \text{the directed arc from origin } i \text{ to breakbulk } k, \\
\bar{t}_j &= \text{the directed arc from breakbulk } l \text{ to destination } j, \text{ and} \\
\bar{u}_{kl} &= \text{the directed arc from breakbulk } k \text{ to breakbulk } l.
\end{align*}
\]
Associated with each arc $\tilde{c}_{ij}, \tilde{s}_i, \tilde{t}_j$ and $\tilde{u}_{kl}$ is a non-negative cost per trailer $c_{ij}, s_i, t_j$ and $u_{kl}$, respectively. Similar to the previous model, we assume $0 \leq a_{ij} < 1$ because a full trailer-load has no need for consolidation.

5.1 Problem Formulation and Notation

To formulate the problem, we define the decision variables

$$x_{ij} = \begin{cases} 1 & \text{if item } (i, j) \text{ shipped from } i, k, l, \text{ to } j, \\ 0 & \text{otherwise,} \end{cases} \quad (i,j) \in (S_k, T_l), \quad k,l \in B$$

$$y_i^s = \text{number of trailers required from } i \text{ to } k, \quad i \in S_k, \quad k \in B$$

$$y_j^t = \text{number of trailers required from } l \text{ to } j, \quad j \in T_l, \quad l \in B$$

$$y_{kl} = \text{number of trailers required from } k \text{ to } l, \quad k,l \in B.$$

A formulation of this problem is

$$\begin{align*}
\text{Min} \quad & z(x, y) = \sum_{(i,j)\in(S,T)} c_{ij}(1-x_{ij}) + \sum_{i\in S} s_i y_i^s + \sum_{j\in T} t_j y_j^t + \sum_{(k,l)\in(B,B)} u_{kl} y_{kl} \\
\text{s.t.} \quad & \sum_{j\in T} a_{ij} x_{ij} \leq y_i^s, \quad i \in S \\
& \sum_{i\in S} a_{ij} x_{ij} \leq y_j^t, \quad j \in T \\
& \sum_{(i,j)\in(S_k,T_l)} a_{ij} x_{ij} \leq y_{kl}, \quad k,l \in B \\
& x_{ij} \in \{0,1\}, \quad (i,j) \in (S,T) \\
& y_i^s, y_j^t, y_{kl} \in \mathbb{Z}_+, \quad (i,j) \in (S_k,T_l), \quad k,l \in B.
\end{align*}$$

While we continue to use the notation in Section 4.1, some modifications and additions are needed because there is more than one breakbulk. Define

$$v_{ij} = u_{kl}, \quad (i,j) \in (S_k,T_l), \quad k,l \in B.$$
Then, we use the following modifications:

\[ G = \{(i,j) \in (S,T) \mid c_{ij} \leq a_{ij}(s_i + v_{ij} + t_j)\} \]

\[ S^h = \{i \in S \mid \sum_{j \in \mathcal{G}} c_{ij} \leq \left[ \sum_{j \in \mathcal{G}} a_{ij} \right] s_i + \sum_{j \in \mathcal{G}} a_{ij}(u_{ij} + t_j)\} \]

\[ T^h = \{j \in T \mid \sum_{i \in \mathcal{G}} c_{ij} \leq \sum_{i \in \mathcal{G}} a_{ij}(s_i + v_{ij}) + \left[ \sum_{i \in \mathcal{G}} a_{ij} \right] t_j\} \]

Further, we introduce the following notation:

\[ F_{kl} = \{(i,j) \in (S_k, T_l) \cap \mathcal{G} \mid i \notin S^h, j \notin T^h\}, \quad k, l \in B \text{ and } k \neq l \]

\[ B^h = \text{the set of breakbulk pairs through which no item is shipped by the heuristic routing} \]

\[ = \{(k,l) \in (B,B) \mid \sum_{(i,j) \in F_{kl}} c_{ij} \leq \sum_{(i,j) \in F_{kl}} a_{ij}(s_i + t_j) + \left[ \sum_{(i,j) \in F_{kl}} a_{ij} \right] u_{kl}\} \]

\[ B^* = \text{the set of breakbulk pairs through which no item is shipped by the optimal routing} \]

\[ = \{(k,l) \in (B,B) \mid x^*_{ij} = 0 \quad \text{for all } (i,j) \in (S_k, T_l)\}. \]

### 5.2 A Heuristic Procedure

Since \( R_c \) is an extension of \( R_s \), it is unary \( \mathcal{NP} \)-hard. To solve \( R_c \) in polynomial time, heuristic procedure HC is developed by extending the procedure HS in Section 4.3. In addition to Step 1 and 2 in HS, HC has an extra step for the detection of expensive arcs between each breakbulk pair.

**Procedure HC**

1. Ship direct items \((i,j) \in G = \{(i,j) \in (S,T) \mid c_{ij} \leq a_{ij}(s_i + v_{ij} + t_j)\}\).

2. For each \( i \in S \), ship items \((i,j), j \in T, \) direct if

\[ \sum_{j \in \mathcal{G}} c_{ij} \leq \left[ \sum_{j \in \mathcal{G}} a_{ij} \right] s_i + \sum_{j \in \mathcal{G}} a_{ij}(v_{ij} + t_j). \]
For each $j \in T$, ship items $(i, j)$, $i \in S$, direct if

$$\sum_{i \in G} c_{ij} \leq \sum_{i \in G} a_{ij}(s_i + v_{ij}) + \left\lceil \sum_{i \in G} a_{ij} \right\rceil t_j.$$

3. For each $(k, l) \in (B, B)$ where $k \neq l$, ship all items $(i, j) \in F_{kl}$ direct if

$$\sum_{(i, j) \in F_{kl}} c_{ij} \leq \sum_{(i, j) \in F_{kl}} a_{ij}(s_i + t_j) + \left\lceil \sum_{(i, j) \in F_{kl}} a_{ij} \right\rceil u_{kl}.$$

4. Ship all other items through the appropriate breakbulks.

Let $n = m_b \times n_b$. Since there are $m_b$ origin and $n_b$ destination terminals, there are at most $n$ items to be delivered, and each step in HC takes $O(n)$ time to perform. Consequently, the time complexity of HC is $O(n)$.

Before we develop error bounds for HC, an example is presented to illustrate the worst-case performance.

**Example 5.1** Consider an instance of $R_c$ with only one item $a_{11}$ and two breakbulks, 1 and 2. Breakbulk 1 has an origin terminal 1, and breakbulk 2 has a destination terminal 1. Let

$$
\begin{align*}
  a_{11} &= \varepsilon \\
  c_{11} &= 1 + 3\varepsilon \\
  s_1 &= 1 \\
  u_{12} &= 1 \\
  t_1 &= 1
\end{align*}
$$

where $0 < \varepsilon < 1$. The solution generated by HC does not ship $a_{11}$ direct since

$$
\begin{align*}
  c_{11} &> 3\varepsilon = a_{11}(s_1 + u_{12} + t_1), \\
  c_{11} &> 1 + 2\varepsilon = \lceil a_{11} \rceil s_1 + a_{11}(u_{12} + t_1), \\
  c_{11} &> 1 + 2\varepsilon = a_{11}(s_1 + u_{12}) + \lceil a_{11} \rceil t_1, \\
  c_{11} &> 1 + 2\varepsilon = a_{11}(s_1 + t_1) + \lceil a_{11} \rceil u_{12}.
\end{align*}
$$
The heuristic solution ships $a_{11}$ through breakbulks with cost of 3 while the optimal solution ships $a_{11}$ direct with the cost of $1 + 3\varepsilon$. Thus, the relative error is

$$\frac{z^h - z^*}{z^*} = \frac{2 - 3\varepsilon}{1 + 3\varepsilon} \to 2 \quad \text{as} \quad \varepsilon \to 0.$$

Next, Lemma 5.1 gives a upper bound for the heuristic solution. Then, an absolute error bound for the solution generated by HC is given in Theorem 5.1.

**Lemma 5.1**

$$z^h \leq \sum_{(i,j) \in G} c_{ij} + \sum_{(i,j) \in G} a_{ij}(s_i + v_{ij} + t_j) + \sum_{i \in S} s_i + \sum_{j \in T} t_j + \sum_{(k,l) \in (B,B)} u_{kl}.$$  

**Proof.** By definition of $R_\varepsilon$,

$$z^h = \sum_{(i,j) \in D^h \cap G} c_{ij} + \sum_{(i,j) \in D^h \cap G} a_{ij} \left[ \sum_{i \in S^h} s_i \right] + \sum_{j \in T^h} \left[ \sum_{i \in S^h} a_{ij} \right] t_j + \sum_{(k,l) \in (S^h, T^h) \cap D^h} \left[ \sum_{(i,j) \in (S^h, T^h) \cap D^h} a_{ij} \right] u_{kl}. \quad (5.1)$$

For the second term on the right hand side of (5.1), observe that

$$\sum_{(i,j) \in D^h \cap G} c_{ij} = \sum_{i \in S^h} \sum_{j \in D^h \cap G} c_{ij} + \sum_{j \in T^h} \sum_{i \in D^h \cap G} c_{ij} - \sum_{(i,j) \in (S^h, T^h) \cap G} c_{ij} + \sum_{(k,l) \in (S^h, T^h) \cap D^h} \sum_{(i,j) \in (S^h, T^h) \cap D^h} c_{ij}. \quad (5.2)$$

Because $\sum_{i \in S} c_{ij} \leq \left[ \sum_{j \in G} a_{ij} \right] s_i + \sum_{j \in G} a_{ij}(v_{ij} + t_j)$ for $i \in S^h$,

$$\sum_{i \in S^h} \sum_{j \in G} c_{ij} \leq \sum_{i \in S^h} \left( \left[ \sum_{j \in G} a_{ij} \right] s_i + \sum_{j \in G} a_{ij}(v_{ij} + t_j) \right) \leq \sum_{i \in S^h} \left( s_i + \sum_{j \in G} a_{ij}(s_i + v_{ij} + t_j) \right). \quad (5.3)$$

Similarly,

$$\sum_{j \in T^h} \sum_{i \in G} c_{ij} \leq \sum_{j \in T^h} \left( \sum_{i \in G} a_{ij}(s_i + v_{ij}) + \left[ \sum_{i \in G} a_{ij} \right] t_j \right)$$
\[
\sum_{(k,l) \in D^h \cap \mathcal{G}} \sum_{(i,j) \in F_{kl}} c_{ij} \leq \sum_{(k,l) \in B^h} \left( \sum_{(i,j) \in F_{kl}} a_{ij}(s_i + v_{ij} + t_j) + \left\lceil \sum_{(i,j) \in F_{kl}} a_{ij} \right\rceil u_{kl} \right) \\
\leq \sum_{(k,l) \in B^h} \left( u_{kl} + \sum_{(i,j) \in F_{kl}} a_{ij}(s_i + v_{ij} + t_j) \right).
\]

Also, because \( c_{ij} > a_{ij}(s_i + v_{ij} + t_j) \) for \((i,j) \in \bar{G} \),

\[
\sum_{(i,j) \in (S^h,T^h) \cap \mathcal{G}} c_{ij} > \sum_{(i,j) \in (S^h,T^h) \cap \mathcal{G}} a_{ij}(s_i + v_{ij} + t_j).
\]

Thus, by (5.2) - (5.6),

\[
\sum_{(i,j) \in D^h \cap \mathcal{G}} c_{ij} \leq \sum_{i \in S^h} \left( s_i + \sum_{j \in \mathcal{G}} a_{ij}(s_i + v_{ij} + t_j) \right) + \sum_{j \in T^h} \left( t_j + \sum_{i \in \mathcal{G}} a_{ij}(s_i + v_{ij} + t_j) \right) \\
- \sum_{(i,j) \in (S^h,T^h) \cap \mathcal{G}} a_{ij}(s_i + v_{ij} + t_j) \\
+ \sum_{(k,l) \in B^h} \left( u_{kl} + \sum_{(i,j) \in F_{kl}} a_{ij}(s_i + v_{ij} + t_j) \right) \\
= \sum_{i \in S^h} s_i + \sum_{j \in T^h} t_j + \sum_{(k,l) \in B^h} u_{kl} + \sum_{(i,j) \in D^h \cap \mathcal{G}} a_{ij}(s_i + v_{ij} + t_j)
\]

where the equality follows from (5.2) with \( c_{ij} \) replaced by \( a_{ij}(s_i + v_{ij} + t_j) \).

For the last three terms on the right hand side of (5.1), because

\[
\sum_{(i,j) \in D^h} a_{ij} = \sum_{i \in S^h} \sum_{j \in D^h} a_{ij} = \sum_{j \in T^h} \sum_{i \in D^h} a_{ij} = \sum_{(k,l) \in B^h} \sum_{(i,j) \in (S^h,T^h) \cap D^h} a_{ij}.
\]

\[
\sum_{i \in S^h} \sum_{j \in D^h} a_{ij} |s_i| + \sum_{j \in T^h} \sum_{i \in D^h} a_{ij} |t_j| + \sum_{(k,l) \in B^h} \sum_{(i,j) \in (S^h,T^h) \cap D^h} a_{ij} |u_{kl}|
\]

\[
\leq \sum_{i \in S^h} \left( 1 + \sum_{j \in D^h} a_{ij} \right) |s_i| + \sum_{j \in T^h} \left( 1 + \sum_{i \in D^h} a_{ij} \right) |t_j| \\
+ \sum_{(k,l) \in B^h} \left( 1 + \sum_{(i,j) \in (S^h,T^h) \cap D^h} a_{ij} \right) |u_{kl}|
\]

\[
\leq \sum_{i \in S^h} |s_i| + \sum_{j \in T^h} |t_j| + \sum_{(k,l) \in B^h} |u_{kl}| + \sum_{(i,j) \in D^h} a_{ij}(s_i + v_{ij} + t_j).
\]
Hence, from equations (5.1), (5.7) and (5.8), because $\bar{D}^h \subseteq \bar{G}$,

$$z^h \leq \sum_{(i,j) \in \bar{G}} c_{ij} + \sum_{(i,j) \in \bar{G}} a_{ij}(s_i + v_{ij} + t_j) + \sum_{i \in S} s_i + \sum_{j \in T} t_j + \sum_{(k,l) \in (B,B)} u_{kl}. \quad \square$$

**Theorem 5.1** The absolute error of the solution generated by HS is no larger than

$$\sum_{i \in S} s_i + \sum_{j \in T} t_j + \sum_{(k,l) \in (B,B)} u_{kl}.$$

**Proof.** The total cost of an optimal routing for $R_c$ is

$$z^* = \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S} [\sum_{j \in D^*} a_{ij}] s_i + \sum_{j \in T} [\sum_{i \in D^*} a_{ij}] t_j + \sum_{(k,l) \in (B,B)} [\sum_{(i,j) \in (S_t, T_t) \cap D^*} a_{ij}] u_{kl}$$

$$\geq \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S} [\sum_{j \in D^*} a_{ij}] s_i + \sum_{j \in T} [\sum_{i \in D^*} a_{ij}] t_j + \sum_{(k,l) \in (B,B)} [\sum_{(i,j) \in (S_t, T_t) \cap D^*} a_{ij}] u_{kl}$$

$$= \sum_{(i,j) \in D^*} c_{ij} + \sum_{(i,j) \in D^*} a_{ij}(s_i + v_{ij} + t_j)$$

$$\geq \sum_{(i,j) \in D^* \cap G} c_{ij} + \sum_{(i,j) \in D^* \cap G} a_{ij}(s_i + v_{ij} + t_j)$$

$$+ \sum_{(i,j) \in D^* \cap \bar{G}} c_{ij} + \sum_{(i,j) \in D^* \cap \bar{G}} a_{ij}(s_i + v_{ij} + t_j)$$

$$= \sum_{(i,j) \in G} c_{ij} + \sum_{(i,j) \in G} a_{ij}(s_i + v_{ij} + t_j)$$

where the last inequality follows from the definitions of $G$ and $\bar{G}$. Using Lemma 5.1, the absolute error of the solution generated by HC is

$$z^h - z^* \leq \sum_{i \in S} s_i + \sum_{j \in T} t_j + \sum_{(k,l) \in (B,B)} u_{kl}. \quad \square$$

In the next two lemmas, we develop a lower bound for the optimal solution and an upper bound for the heuristic solution generated by HC. Then, Theorem 5.2 gives a relative error bound for HC.

Observe that Lemma 4.2 also holds for HC. Consequently, without loss of generality, we assume that $D^* \cap G = \emptyset$. Note that if $D^* \cap G = \emptyset$, then $D^* \cap \bar{G} = D^*$ and $\bar{D}^* \cap G = G$. 

53
Lemma 5.2

\[
2z^* > \sum_{(i,j) \in D} c_{ij} + \sum_{i \in S^*} \sum_{j \in G} c_{ij} + \sum_{j \in T^* \cup T^h} \sum_{i \in G} c_{ij} \\
+ \sum_{i \in S^* \cup S^h} \left( s_i + \sum_{j \in G} a_{ij}(t_j + v_{ij}) \right) + \sum_{j \in T^* \cup T^h} \left( t_j + \sum_{i \in G} a_{ij}(s_i + v_{ij}) \right).
\]

Proof. The total shipping cost for an optimal solution is

\[
z^* = \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S^*} \sum_{j \in D^*} a_{ij} s_i + \sum_{j \in T^*} \sum_{i \in D^*} a_{ij} t_j + \sum_{(k,l) \in D^*} \sum_{(i,j) \in (S_k,T_l) \cap D^*} a_{ij} u_{kl} \\
\geq \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S^*} s_i + \sum_{j \in D^*} a_{ij} t_j + \sum_{(i,j) \in D^*} a_{ij} u_{ij} \\
= \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S^*} s_i + \sum_{j \in D^*} a_{ij} t_j + \sum_{(i,j) \in D^*} a_{ij} u_{ij} \\
= \sum_{(i,j) \in D^* \cap G} c_{ij} + \sum_{i \in S^*} s_i + \sum_{j \in D^* \cap G} a_{ij} (t_j + v_{ij}) + \sum_{(i,j) \in D^* \cap G} a_{ij} (t_j + v_{ij}) \\
= \sum_{(i,j) \in D^* \cap G} c_{ij} + \sum_{i \in S^*} s_i + \sum_{j \in D^* \cap G} a_{ij} (t_j + v_{ij}) + \sum_{i \in S^*} \sum_{j \in D^* \cap G} a_{ij} (t_j + v_{ij}).
\]

Since \( S^* \subseteq D^*(\cdot,j) \) for each \( j \in N \), observe that

\[
\sum_{(i,j) \in D^* \cap G} c_{ij} = \sum_{i \in S^* \cap S^h} \sum_{j \in G} c_{ij} + \sum_{i \in S^* \cap S^h} \sum_{j \in G} c_{ij} + \sum_{i \in S^* \cap D^* \cap G} \sum_{j \in D^* \cap G} c_{ij} \\
> \sum_{i \in S^* \cap S^h} \sum_{j \in G} c_{ij} + \sum_{i \in S^* \cap S^h} \sum_{j \in G} a_{ij} \left( \sum_{j \in G} s_i + \sum_{j \in G} a_{ij} (t_j + v_{ij}) \right) \\
+ \sum_{i \in S^* \cap D^* \cap G} \sum_{j \in D^* \cap G} a_{ij} (s_i + v_{ij} + t_j) \\
\geq \sum_{i \in S^* \cap S^h} \sum_{j \in G} c_{ij} + \sum_{i \in S^* \cap S^h} \sum_{j \in G} a_{ij} (t_j + v_{ij}) \\
+ \sum_{i \in S^* \cap D^* \cap G} \sum_{j \in D^* \cap G} a_{ij} (t_j + v_{ij}).
\]  

On the right hand side of equation (5.9), the second term follows from the definition of \( S^h \), and the third term follows from the definition of \( \bar{G} \). Hence,

\[
z^* > \sum_{i \in S^* \cap S^h} \sum_{j \in G} c_{ij} + \sum_{i \in S^* \cap S^h} \sum_{j \in G} a_{ij} (t_j + v_{ij}) + \sum_{i \in S^* \cap D^* \cap G} \sum_{j \in G} a_{ij} (t_j + v_{ij})
\]
\[
\sum_{i \in D^*} s_i + \sum_{(i,j) \in G} a_{ij}(t_j + v_{ij}) + \sum_{i \in D^*} \sum_{j \in D \cap \mathcal{G}} a_{ij}(t_j + v_{ij}) \\
= \sum_{i \in D^* \cup \mathcal{G}} \sum_{j \in \mathcal{G}} c_{ij} + \sum_{i \in D^* \cup \mathcal{G}} \left( s_i + \sum_{j \in \mathcal{G}} a_{ij}(t_j + v_{ij}) \right) + \sum_{(i,j) \in G} a_{ij}(t_j + v_{ij}).
\]

By a similar argument,
\[
z^* > \sum_{j \in \mathcal{T^* \cup \mathcal{T}^h}} \sum_{i \in \mathcal{G}} c_{ij} + \sum_{j \in \mathcal{T^* \cup \mathcal{T}^h}} \left( t_j + \sum_{i \in \mathcal{G}} a_{ij}(s_i + v_{ij}) \right) + \sum_{(i,j) \in G} a_{ij}(s_i + v_{ij}).
\]

Consequently,
\[
2z^* > \sum_{i \in D^* \cup \mathcal{G}} \sum_{j \in \mathcal{G}} c_{ij} + \sum_{j \in \mathcal{T^* \cup \mathcal{T}^h}} \sum_{i \in \mathcal{G}} c_{ij} + \sum_{(i,j) \in G} a_{ij}(s_i + v_{ij} + t_j) \\
+ \sum_{i \in D^* \cup \mathcal{G}} \left( s_i + \sum_{j \in \mathcal{G}} a_{ij}(t_j + v_{ij}) \right) + \sum_{j \in \mathcal{T^* \cup \mathcal{T}^h}} \left( t_j + \sum_{i \in \mathcal{G}} a_{ij}(s_i + v_{ij}) \right) \\
\geq \sum_{(i,j) \in G} c_{ij} + \sum_{i \in D^* \cup \mathcal{G}} \sum_{j \in \mathcal{G}} c_{ij} + \sum_{j \in \mathcal{T^* \cup \mathcal{T}^h}} \sum_{i \in \mathcal{G}} c_{ij} \\
+ \sum_{i \in D^* \cup \mathcal{G}} \left( s_i + \sum_{j \in \mathcal{G}} a_{ij}(t_j + v_{ij}) \right) + \sum_{j \in \mathcal{T^* \cup \mathcal{T}^h}} \left( t_j + \sum_{i \in \mathcal{G}} a_{ij}(s_i + v_{ij}) \right). \quad \Box
\]

Lemma 5.3 \( z^* \geq \sum_{(k,l) \in \mathcal{B} \setminus \mathcal{G}} \sum_{(i,j) \in (S_k, T_l)} c_{ij} + \sum_{(k,l) \in \mathcal{B} \setminus \mathcal{G}} u_{kl}. \)

Proof. For each \((k,l) \in \mathcal{B}^*\), all items \((i,j) \in (S_k, T_l)\) are shipped direct, which implies that \(D^* \cap \mathcal{G} \cap (S_k, T_l) = \mathcal{G} \cap (S_k, T_l)\). Hence, the total shipping cost for an optimal solution is

\[
z^* = \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in D^* \cup \mathcal{G}} \left[ \sum_{j \in D^*} a_{ij} \right] s_i + \sum_{j \in \mathcal{T^*} \cup \mathcal{T}^h} \left[ \sum_{i \in D^*} a_{ij} \right] t_j + \sum_{(k,l) \in \mathcal{B}^* \setminus \mathcal{G}} \sum_{(i,j) \in (S_k, T_l) \cap D^*} a_{ij} u_{kl} \\
\geq \sum_{(i,j) \in D^* \cap \mathcal{G}} c_{ij} + \sum_{(k,l) \in \mathcal{B}^* \setminus \mathcal{G}} u_{kl} \\
= \sum_{(k,l) \in \mathcal{B} \setminus \mathcal{G}} \sum_{(i,j) \in (S_k, T_l) \cap D^*} c_{ij} + \sum_{(k,l) \in \mathcal{B} \setminus \mathcal{G}} u_{kl} \\
\geq \sum_{(k,l) \in \mathcal{B} \setminus \mathcal{G}} \sum_{(i,j) \in (S_k, T_l) \cap D^*} c_{ij} + \sum_{(k,l) \in \mathcal{B} \setminus \mathcal{G}} u_{kl}. \quad (5.10)
\]
Inequality (5.10) follows because \( \bar{D}^h \subseteq \bar{G} \). For each \((k, l) \in (B, B)\), if \((S_k, T_l) \cap \bar{D}^h \neq \emptyset\), then some item \((i, j) \in (S_k, T_l)\) is shipped on arc \( \bar{u}_{kl} \) in the heuristic solution. Thus, by Step 3 of HC,

\[
\sum_{(k,l) \in B^* \cap \bar{B}^h} \sum_{(i,j) \in (S_k, T_l) \cap \bar{D}^h} c_{ij} > \sum_{(k,l) \in B^* \cap \bar{B}^h} \left[ \sum_{(i,j) \in (S_k, T_l) \cap \bar{D}^h} a_{ij} \right] \bar{u}_{kl} \geq \sum_{(k,l) \in B^* \cap \bar{B}^h} \bar{u}_{kl}.
\]

Therefore, from (5.10),

\[
z^* > \sum_{(k,l) \in B^* \cap \bar{B}^h} \sum_{(i,j) \in (S_k, T_l) \cap \bar{G}} c_{ij} + \sum_{(k,l) \in B^* \cap \bar{B}^h} \sum_{(i,j) \in (S_k, T_l) \cap \bar{G}} \bar{u}_{kl} + \sum_{(k,l) \in B^* \cap \bar{B}^h} \bar{u}_{kl} \quad \square
\]

**Lemma 5.4**

\[
z^h \leq \sum_{(i,j) \in \bar{G}} c_{ij} + \sum_{i \in S^h \cap \bar{B}^h} \sum_{j \in \bar{G}} c_{ij} + \sum_{j \in T^h \cap \bar{B}^h} \sum_{i \in \bar{G}} c_{ij} + \sum_{(k,l) \in B^* \cap \bar{B}^h} \sum_{(i,j) \in (S_k, T_l) \cap \bar{G}} c_{ij} + \sum_{(k,l) \in B^* \cap \bar{B}^h} \sum_{(i,j) \in (S_k, T_l) \cap \bar{G}} \bar{u}_{kl} + \sum_{i \in S^h \cap \bar{B}^h} \sum_{j \in \bar{T}^h \cap \bar{B}^h} a_{ij}(t_j + v_{ij}) + \sum_{j \in T^h \cap \bar{B}^h} \sum_{i \in \bar{G}} a_{ij}(s_i + v_{ij}).
\]

**Proof.** By expanding equation (5.2) and rearranging the terms,

\[
\sum_{(i,j) \in \bar{D}^h \cap \bar{G}} c_{ij} = \sum_{i \in S^h \cap \bar{B}^h} \sum_{j \in \bar{G}} c_{ij} + \sum_{j \in T^h \cap \bar{B}^h} \sum_{i \in \bar{G}} c_{ij} + \sum_{(k,l) \in B^* \cap \bar{B}^h} \sum_{(i,j) \in F_{kl}} c_{ij} + \sum_{(k,l) \in B^* \cap \bar{B}^h} \sum_{(i,j) \in F_{kl}} \bar{u}_{kl} + \sum_{i \in S^h \cap \bar{B}^h} \sum_{j \in \bar{T}^h \cap \bar{B}^h} a_{ij}(s_i + v_{ij} + t_j) + \sum_{j \in T^h \cap \bar{B}^h} \sum_{i \in \bar{G}} a_{ij}(s_i + v_{ij} + t_j). \quad (5.11)
\]

By the same argument as in (5.3), (5.4) and (5.5),

\[
\sum_{i \in S^h \cap \bar{B}^h} \sum_{j \in \bar{G}} c_{ij} \leq \sum_{i \in S^h \cap \bar{B}^h} \left( s_i + \sum_{j \in \bar{G}} a_{ij}(s_i + v_{ij} + t_j) \right), \quad (5.12)
\]

\[
\sum_{j \in T^h \cap \bar{B}^h} \sum_{i \in \bar{G}} c_{ij} \leq \sum_{j \in T^h \cap \bar{B}^h} \left( t_j + \sum_{i \in \bar{G}} a_{ij}(s_i + v_{ij} + t_j) \right), \quad (5.13)
\]

\[
\sum_{(k,l) \in B^* \cap \bar{B}^h} \sum_{(i,j) \in F_{kl}} c_{ij} \leq \sum_{(k,l) \in B^* \cap \bar{B}^h} \left( \bar{u}_{kl} + \sum_{(i,j) \in F_{kl}} a_{ij}(s_i + v_{ij} + t_j) \right). \quad (5.14)
\]
From (5.11), (5.12), (5.13) and (5.14),

\[
\sum_{(i,j) \in D^h \cap G} c_{ij} \leq \sum_{i \in S^h} \sum_{j \in G} c_{ij} + \sum_{j \in T^h} \sum_{i \in G} c_{ij} + \sum_{(k,l) \in E^h \cap G} \sum_{(i,j) \in F_{kl}} c_{ij} \\
+ \sum_{i \in S^h} \left( s_i + \sum_{j \in G} a_{ij}(s_i + v_{ij} + t_j) \right) \\
+ \sum_{j \in T^h} \left( t_j + \sum_{i \in G} a_{ij}(s_i + v_{ij} + t_j) \right) \\
- \sum_{(i,j) \in (S^h, T^h) \cap G} a_{ij}(s_i + v_{ij} + t_j) \\
+ \sum_{(k,l) \in E^h \cap G} \left( u_{kl} + \sum_{(i,j) \in F_{kl}} a_{ij}(s_i + v_{ij} + t_j) \right)
\]

\[
= \sum_{i \in S^h} \sum_{j \in G} c_{ij} + \sum_{j \in T^h} \sum_{i \in G} c_{ij} + \sum_{(k,l) \in E^h \cap G} \sum_{(i,j) \in F_{kl}} c_{ij} \\
+ \sum_{i \in S^h} s_i + \sum_{j \in T^h} t_j + \sum_{(k,l) \in E^h \cap G} u_{kl} \\
+ \sum_{(i,j) \in D^h \cap G} a_{ij}(s_i + v_{ij} + t_j) - \sum_{i \in S^h} \sum_{j \in G} a_{ij}(s_i + v_{ij} + t_j) \\
- \sum_{j \in T^h} \sum_{i \in G} a_{ij}(s_i + v_{ij} + t_j) - \sum_{(k,l) \in E^h \cap G} \sum_{(i,j) \in F_{kl}} a_{ij}(s_i + v_{ij} + t_j) \\
- \sum_{(k,l) \in E^h \cap G} \sum_{(i,j) \in F_{kl}} a_{ij}(s_i + v_{ij} + t_j)
\]

(5.15)

where the last four terms in the equality are obtained from (5.11) with \( c_{ij} \) replaced by \( a_{ij}(s_i + v_{ij} + t_j) \). Hence, from equations (5.1), (5.15) and (5.8),

\[
z^h \leq \sum_{(i,j) \in E^h \cap G} c_{ij} + \sum_{i \in S^h} \sum_{j \in G} c_{ij} + \sum_{j \in T^h} \sum_{i \in G} c_{ij} + \sum_{(k,l) \in E^h \cap G} \sum_{(i,j) \in F_{kl}} c_{ij} \\
+ \sum_{i \in S^h} s_i + \sum_{j \in T^h} t_j + \sum_{(k,l) \in E^h \cap G} u_{kl} \\
+ \sum_{(i,j) \in D^h \cap G} a_{ij}(s_i + v_{ij} + t_j) - \sum_{i \in S^h} \sum_{j \in G} a_{ij}(s_i + v_{ij} + t_j) \\
- \sum_{j \in T^h} \sum_{i \in G} a_{ij}(s_i + v_{ij} + t_j) - \sum_{(k,l) \in E^h \cap G} \sum_{(i,j) \in F_{kl}} a_{ij}(s_i + v_{ij} + t_j) \\
+ \sum_{i \in S^h} s_i + \sum_{j \in T^h} t_j + \sum_{(k,l) \in E^h \cap G} u_{kl} + \sum_{(i,j) \in D^h \cap G} a_{ij}(s_i + v_{ij} + t_j)
\]

\[
\leq \sum_{(i,j) \in E^h \cap G} c_{ij} + \sum_{i \in S^h} \sum_{j \in G} c_{ij} + \sum_{j \in T^h} \sum_{i \in G} c_{ij} + \sum_{(k,l) \in E^h \cap G} \sum_{(i,j) \in (S_0, T_1) \cap G} c_{ij}
\]

57
\[ + \sum_{i \in S^* U^h} s_i + \sum_{j \in T^* U^h} t_j + \sum_{(k,l) \in B^* U^h} u_{kl} \\
+ \sum_{(i,j) \in \tilde{G}} a_{ij}(s_i + v_{ij} + t_j) - \sum_{(i,j) \in \tilde{G}} \sum_{j \in \tilde{G}} a_{ij}(s_i + v_{ij} + t_j) \\
- \sum_{j \in T^* T^h} \sum_{i \in \tilde{G}} a_{ij}(s_i + v_{ij} + t_j) - \sum_{(k,l) \in B^* T^h} \sum_{(i,j) \in \tilde{F}_{kl}} a_{ij}(s_i + v_{ij} + t_j). \tag{5.16} \]

The inequality (5.16) follows because \( F_{kl} \subseteq \tilde{G} \) and \( D^h \subseteq \tilde{G} \). Since the last four terms on the right hand side of (5.16)

\[
\sum_{(i,j) \in \tilde{G}} a_{ij}(s_i + v_{ij} + t_j) - \sum_{i \in S^* T^h} \sum_{j \in \tilde{G}} a_{ij}(s_i + v_{ij} + t_j) \\
- \sum_{j \in T^* T^h} \sum_{i \in \tilde{G}} a_{ij}(s_i + v_{ij} + t_j) - \sum_{(k,l) \in B^* T^h} \sum_{(i,j) \in \tilde{F}_{kl}} a_{ij}(s_i + v_{ij} + t_j) \\
\leq \sum_{(i,j) \in \tilde{G}} a_{ij}(t_j + v_{ij}) - \sum_{i \in S^* T^h} \sum_{j \in \tilde{G}} a_{ij}(s_i + v_{ij} + t_j) \\
+ \sum_{(i,j) \in \tilde{G}} a_{ij}(s_i + v_{ij}) - \sum_{j \in T^* T^h} \sum_{i \in \tilde{G}} a_{ij}(s_i + v_{ij} + t_j) \\
\leq \sum_{i \in S^* T^h} \sum_{j \in \tilde{G}} a_{ij}(t_j + v_{ij}) + \sum_{j \in T^* T^h} \sum_{i \in \tilde{G}} a_{ij}(s_i + v_{ij}),
\]

we obtain

\[ z^h \leq \sum_{(i,j) \in \tilde{G}} c_{ij} + \sum_{i \in S^* T^h} \sum_{j \in \tilde{G}} c_{ij} + \sum_{j \in T^* T^h} \sum_{i \in \tilde{G}} c_{ij} + \sum_{(k,l) \in B^* T^h} \sum_{(i,j) \in (S_k, T_i)} c_{ij} \\
+ \sum_{i \in S^* U^h} s_i + \sum_{j \in T^* U^h} t_j + \sum_{(k,l) \in B^* U^h} u_{kl} \\
+ \sum_{i \in S^* U^h} \sum_{j \in \tilde{G}} a_{ij}(t_j + v_{ij}) + \sum_{j \in T^* U^h} \sum_{i \in \tilde{G}} a_{ij}(s_i + v_{ij}). \quad \square \]

**Theorem 5.2** The relative error of the solution generated by HC is 2. Further, the bound is tight.

**Proof.** Combining Lemma 5.2 and 5.3, we obtain

\[ 3z^* \geq \sum_{(i,j) \in \tilde{G}} c_{ij} + \sum_{i \in S^* T^h} \sum_{j \in \tilde{G}} c_{ij} + \sum_{j \in T^* T^h} \sum_{i \in \tilde{G}} c_{ij} + \sum_{(k,l) \in B^* T^h} \sum_{(i,j) \in (S_k, T_i)} c_{ij} \\
+ \sum_{i \in S^* U^h} s_i + \sum_{j \in T^* U^h} t_j + \sum_{(k,l) \in B^* U^h} u_{kl} \\
+ \sum_{i \in S^* U^h} \sum_{j \in \tilde{G}} a_{ij}(t_j + v_{ij}) + \sum_{j \in T^* U^h} \sum_{i \in \tilde{G}} a_{ij}(s_i + v_{ij}). \]

58
From Lemma 5.4, we have $3z^* > z^h$. This implies that

$$\frac{z^h - z^*}{z^*} \leq 2.$$ 

By Example 5.1, the bound is tight. □

5.3 Conclusion

We extend the single breakbulk problem into a clique breakbulk problem where an arc exists between every breakbulk pair. In practice, it is common to have clique breakbulks since there is usually enough volume to send items directly between primary breakbulks without going through an intermediate breakbulk.

We develop an $O(n)$ time heuristic procedure, HC, where $n$ is the total number of items. The absolute error for HC is equal to the total cost of dispatching one trailer on each arc going to, from and between breakbulks. Also, we show that the relative error for HS is two.
Figure 5.1: Clique breakbulk delivery network with 4 breakbulks
CHAPTER 6

MULTIPLE BREAKBULK PROBLEM

In this chapter, we study a general freight delivery systems with multiple breakbulks where breakbulks may not form a clique. Each item can be shipped either direct or on any path through the breakbulks. We denote this problem as \( R_m \).

Problem \( R_m \) is unary \( \mathcal{NP} \)-hard since it is a generalization of \( R_c \). We propose a heuristic procedure which first finds a shortest path for each breakbulk pair in \( R_m \). Then, using the costs for the shortest paths, we convert the \( R_m \) to a clique breakbulk problem \( R_c \). Next, we find a solution to \( R_c \) with procedure HC. Finally, the solution for \( R_c \) is converted to a feasible solution for \( R_m \).

6.1 Problem Formulation and Notation

Similar to \( R_c \) in Chapter 5, we have a set of breakbulk terminals \( k \in B = \{1, 2, \ldots, b\} \). For each \( k \in B \), there are a set of origin terminals \( i \in S_k = \{m_{k-1} + 1, m_{k-1} + 2, \ldots, m_k\} \), and a set of destination terminals \( j \in T_k = \{n_{k-1} + 1, n_{k-1} + 2, \ldots, n_k\} \), where \( m_0 = n_0 = 0 \). Let \( S = \cup_{k \in B} S_k \) and \( T = \cup_{l \in B} T_l \). Thus, \( |S| = m_b \) and \( |T| = n_b \). Each item \((i, j)\) has size \( a_{ij} \) where \( 0 \leq a_{ij} < 1 \).
To formulate the problem, we define \( A \) be the set of arcs between breakbulk terminals. The decision variables are

\[
x_{ij} = \begin{cases} 
0 & \text{if item } (i,j) \text{ shipped directly,} \\
1 & \text{otherwise,} 
\end{cases} \quad (i,j) \in (S,T), \; k,l \in B
\]

\[
y_i^k = \text{number of trailers required from } i \text{ to } k, \quad i \in S_k, \; k \in B
\]

\[
y_j^l = \text{number of trailers required from } l \text{ to } j, \quad j \in T_l, \; l \in B
\]

\[
y_{qr} = \text{number of trailers required from } q \text{ to } r, \quad (q,r) \in A
\]

\[
d_{ijqr} = \begin{cases} 
1 & \text{if item } (i,j) \text{ shipped from } q \text{ to } r, \\
0 & \text{otherwise,} 
\end{cases} \quad (i,j) \in (S,T), \; (q,r) \in A.
\]

A formulation of this problem is

\[
\begin{aligned}
\text{Min} & \quad z(x,y) = \sum_{(i,j) \in (S,T)} c_{ij}(1 - x_{ij}) + \sum_{i \in S} s_i y_i^k + \sum_{j \in T} t_j y_j^l + \sum_{(q,r) \in A} u_{kl} y_{kl} \\
\text{s. t.} & \quad x_{ij} = \sum_{r \in \{e \in B \mid (k,e) \in A\}} d_{ijkr}, \quad (i,j) \in (S_k,T), \; k \in B \\
& \quad x_{ij} = \sum_{r \in \{e \in B \mid (e,l) \in A\}} d_{ijrl}, \quad (i,j) \in (S,T_l), \; l \in B \\
& \quad \sum_{r \in \{e \in B \mid (e,q) \in A\}} d_{ijqr} = \sum_{r \in \{e \in B \mid (e,r) \in A\}} d_{ijqr}, \quad (i,j) \in (S,T), \; q \in B \setminus \{k,l\} \\
& \quad \sum_{j \in T} a_{ij} x_{ij} \leq y_i^k, \quad i \in S \\
& \quad \sum_{i \in S} a_{ij} x_{ij} \leq y_j^l, \quad j \in T \\
& \quad \sum_{(i,j) \in (S,T)} a_{ij} d_{ijqr} \leq y_{qr}, \quad (q,r) \in A \\
& \quad x_{ij}, \; d_{ijqr} \in \{0,1\}, \quad (i,j) \in (S,T), \; (q,r) \in A \\
& \quad y_i^k, \; y_j^l, \; y_{qr} \in \mathbb{Z}_+, \quad (i,j) \in (S,T), \; (q,r) \in A.
\end{aligned}
\]

We continue to use the notation in Chapter 5. In addition, for each breakbulk pair \((k,l) \in (B,B)\), denote \( P_{kl} \) as the path found by the shortest path algorithm. Suppose \( P_{kl} = (k,k_1,k_2,\ldots,k_r,l) \) where \( k_1,k_2,\ldots,k_r \in B \) and \( r \leq b - 2 \). Define the cost per trailer for path \( P_{kl} \) to be

\[
w_{kl} = u_{kk_1} + \sum_{i=1}^{r-1} u_{k_ik_{i+1}} + u_{k_r l},
\]

62
and let
\[ \hat{v}_{ij} = w_{kl}, \quad (i, j) \in (S_k, T_l), \quad k, l \in B. \]

6.2 A Heuristic Procedure

Since \( R_m \) is unary \( \mathcal{NP} \)-hard, we develop a heuristic which utilizes procedure HC to solve \( R_m \).

**Procedure HM**

1. Use a shortest path algorithm to find a path for each breakbulk pair in \( R_m \).
2. Construct a clique breakbulk problem \( R_c \) by creating an arc \( \tilde{w}_{kl} \) with the cost per trailer \( w_{kl} \) for each \( (k, l) \in (B, B) \) in \( R_m \).
3. Use procedure HC to solve the clique breakbulk problem constructed in Step 2.
4. Ship directly each item that is shipped directly in \( R_c \), and ship the remaining items through the shortest paths found in Step 1.

A shortest path algorithm such as Dijkstra’s Algorithm can be used to find a shortest path in \( O(b^2) \) time for each breakbulk pair. Therefore, Step 1 can be completed in \( O(b^4) \) time. Step 2 takes \( O(b^3) \) time to compute the cost per trailer for each arc constructed in Step 1, and HC solves \( R_c \) in \( O(n) \) time. Consequently, HM runs in \( O(b^4 + n) \) time, where \( n \) is the maximum number of items to be shipped through the network.

Next, we perform a worst-case analysis to HM. An example is presented next to illustrate the worst-case scenario for HM.
Next, we perform a worst-case analysis to HM. An example is presented next to illustrate the worst-case scenario for HM.

**Example 6.1** Consider an instance of $R_m$ that has $b - 1$ origin, $b$ destination and $b$ breakbulk terminals as shown in Figure 6.1. Each breakbulk except for 1 and $b$ has one origin and one destination terminal. Breakbulk 1 only has an origin terminal. Breakbulk $b$ does not have an origin but has two destination terminals, $b$ and $b + 1$. Let

![Diagram showing worst-case example for $R_m$.](image)

Figure 6.1: Worst-case example for $R_m$. 

64
\[ a_{ij} = \varepsilon, \quad i = 1, 2, \ldots, b-1; \ j = 2, 3, \ldots, b; \ i \neq j, \ (i, j) \neq (1, b) \]
\[ a_{1,b} = 1 - \varepsilon(b-2)^2 \]
\[ a_{1,b+1} = \varepsilon((b-2)^2 + 1) \]
\[ c_{ij} = 1, \quad i = 1, 2, \ldots, b-1; \ j = 2, 3, \ldots, b; \ i \neq j, \ (i, j) \neq (1, b) \]
\[ c_{1,b} = 2 \]
\[ c_{1,b+1} = \varepsilon((b-2)^2 + 2) \]
\[ s_i = 0, \quad i = 1, 2, \ldots, b-1 \]
\[ t_j = 0, \quad j = 2, 3, \ldots, b+1 \]
\[ u_{1b} = 1 \]
\[ u_{bj} = \varepsilon, \quad j = 2, 3, \ldots, b-1 \]
\[ u_{i1} = \varepsilon, \quad i = 2, 3, \ldots, b-1 \]

where \( 0 < \varepsilon \leq 1/b^2 \) and item \((i, j) \in (S_i, T_j)\). Thus, the shortest paths between breakbulks are

\[ P_{ib} = \{1, b\} \]
\[ P_{ij} = \{1, b, j\}, \quad j = 2, 3, \ldots, b-1 \]
\[ P_{ib} = \{i, 1, b\}, \quad i = 2, 3, \ldots, b-1 \]
\[ P_{ij} = \{i, 1, b, j\}, \quad i = 2, 3, \ldots, b-1; \ j = 2, 3, \ldots, b-1; \ i \neq j. \]

For arcs constructed in Step 2 of HM, the costs per trailer are

\[ w_{1b} = 1, \]
\[ w_{1j} = 1 + \varepsilon, \quad j = 2, 3, \ldots, b-1, \]
\[ w_{ib} = 1 + \varepsilon, \quad i = 2, 3, \ldots, b-1, \]
\[ w_{ij} = 1 + 2\varepsilon, \quad i = 2, 3, \ldots, b-1; \ j = 2, 3, \ldots, b-1; \ i \neq j. \]

Observe that, for each \( i = 1, 2, \ldots, b-1, \ j = 2, 3, \ldots, b \) and \( i \neq j, \ (i, j) \neq (1, b) \), path \( P_{ij} = \{i, 1, b, j\} \) has no more than one item \((i, j)\) going through it. The heuristic routing ships \((b-1)(b-2)\) items \((i, j)\) for \( i = 1, 2, \ldots, b-1; \ j = 2, 3, \ldots, b; \ i \neq j, \ (i, j) \neq (1, b) \) directly because

\[ c_{ij} \leq w_{ij}. \]

Items \((1, b)\) and \((1, b+1)\) are shipped through the breakbulk because

\[ c_{1b} = 2 > 1 - \varepsilon(b-2)^2 = a_{1b}(s_1 + w_{1b} + t_b) \]
\[ c_{1,b+1} = \varepsilon((b-2)^2 + 2) > \varepsilon((b-2)^2 + 1) = a_{1,b+1}(s_1 + w_{1b} + t_{b+1}) \]
\[ c_{1b} + c_{1,b+1} = 2 + \varepsilon((b-2)^2 + 2) > 2 = [a_{1b} + a_{1,b+1}]w_{1b}. \]
Thus, the cost of the heuristic routing is
\[ z^h = (b - 1)(b - 2) + 2. \]

However, the optimal routing ships item \((1, b+1)\) directly, and all other items through the breakbulks with cost
\[
z^* = c_{1,b+1} + u_{1b} + \sum_{j=2}^{b-1} u_{bj} + \sum_{i=2}^{b-1} u_{il} \\
= \varepsilon((b - 2)^2 + 2) + 1 + 2(b - 2)\varepsilon \to 1 \quad \text{as } \varepsilon \to 0.
\]

Therefore, in this example,
\[
\frac{(z^h - z^*)}{z^*} \to (b - 1)(b - 2) + 1 \quad \text{as } \varepsilon \to 0.
\]

In the following lemmas, we develop lower bounds for the optimal solution and an upper bound for the heuristic solution generated by HM. Then, Theorem 6.2 gives a relative error bound for HM. Recall from Lemma 4.2 that we assume \(D^* \cap G = \emptyset\), which implies that \(G \subseteq \bar{D}^*, D^* \cap \bar{G} = D^*\) and \(\bar{D}^* \cap G = G\).

**Lemma 6.1**
\[
z^* > \sum_{i \in S^*} \sum_{j \in \bar{G}} c_{ij} + \sum_{i \in S^*} \sum_{j \in \bar{G}} \left( s_i + \sum_{j \in \bar{G}} a_{ij}(t_j + \hat{v}_{ij}) \right) + \sum_{(i,j) \in G} a_{ij}(t_j + \hat{v}_{ij}).
\]

**Proof.** The total shipping cost for an optimal solution is
\[
z^* = \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S^*} \sum_{j \in D^*} a_{ij}s_i + \sum_{j \in T^*} \sum_{i \in D^*} a_{ij}t_j \\
+ \sum_{(k,l) \in (B,B)} \sum_{(i,j) \in (S_k,T_l) \cap D^*} a_{ij} \sum_{(p,q) \in P_{kl}^*} u_{pq} \\
\geq \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S^*} s_i + \sum_{j \in T^*} \sum_{i \in D^*} a_{ij}t_j + \sum_{(k,l) \in (B,B)} \sum_{(i,j) \in (S_k,T_l) \cap D^*} a_{ij}w_{kl} \\
= \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S^*} s_i + \sum_{(i,j) \in D^*} a_{ij}t_j + \sum_{(i,j) \in D^*} a_{ij} \hat{v}_{ij} \\
= \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S^*} s_i + \sum_{(i,j) \in D^*} a_{ij}(t_j + \hat{v}_{ij}) + \sum_{(i,j) \in D^*} a_{ij}(t_j + \hat{v}_{ij}) \\
= \sum_{(i,j) \in D^* \cap G} c_{ij} + \sum_{i \in S^*} s_i + \sum_{(i,j) \in D^* \cap G} a_{ij}(t_j + \hat{v}_{ij}) + \sum_{(i,j) \in D^* \cap G} a_{ij}(t_j + \hat{v}_{ij}).
\]

66
Since $S^* \subseteq D^*(\cdot, j)$ for each $j \in N$, observe that

\[
\sum_{(i,j) \in D^* \cap \mathcal{G}} c_{ij} = \sum_{i \in S^* \cap \mathcal{S}^h} \sum_{j \in \mathcal{G}} c_{ij} + \sum_{i \in S^* \cap \mathcal{S}^h} \sum_{j \in \mathcal{G}} c_{ij} + \sum_{i \in S^* \cap \mathcal{D} \cap \mathcal{G}} c_{ij} > \sum_{i \in S^* \cap \mathcal{S}^h} \sum_{j \in \mathcal{G}} c_{ij} + \sum_{i \in S^* \cap \mathcal{S}^h} \left( \sum_{j \in \mathcal{G}} a_{ij} s_i + \sum_{j \in \mathcal{G}} a_{ij} (t_j + \hat{v}_{ij}) \right) + \sum_{i \in S^* \cap \mathcal{D} \cap \mathcal{G}} a_{ij} (s_i + \hat{v}_{ij} + t_j) \tag{6.1}
\]

\[
\geq \sum_{i \in S^* \cap \mathcal{S}^h} \sum_{j \in \mathcal{G}} c_{ij} + \sum_{i \in S^* \cap \mathcal{S}^h} \left( s_i + \sum_{j \in \mathcal{G}} a_{ij} (t_j + \hat{v}_{ij}) \right) + \sum_{i \in S^* \cap \mathcal{D} \cap \mathcal{G}} a_{ij} (t_j + \hat{v}_{ij}). \tag{6.2}
\]

On the right hand side of equation (6.1), the second term follows from the definition of $\tilde{S}^h$, and the third term follows from the definition of $\tilde{G}$. Hence,

\[
z^* > \sum_{i \in S^* \cap \mathcal{S}^h} \sum_{j \in \mathcal{G}} c_{ij} + \sum_{i \in S^* \cap \mathcal{S}^h} \left( s_i + \sum_{j \in \mathcal{G}} a_{ij} (t_j + \hat{v}_{ij}) \right) + \sum_{i \in S^* \cap \mathcal{D} \cap \mathcal{G}} a_{ij} (t_j + \hat{v}_{ij})
+ \sum_{i \in S^* \cap \mathcal{S}^h} \left( s_i + \sum_{j \in \mathcal{G}} a_{ij} (t_j + \hat{v}_{ij}) \right) + \sum_{(i,j) \in \mathcal{G}} a_{ij} (t_j + \hat{v}_{ij}). \quad \Box
\]

**Lemma 6.2**: For $b \geq 3,$

\[
(b-1)(b-2)+1 z^* > \sum_{j \in T^* \cap \mathcal{T}^h} \sum_{i \in \mathcal{G}} c_{ij} + \sum_{j \in T^* \cup \mathcal{T}^h} \left( t_j + \sum_{i \in \mathcal{G}} a_{ij} s_i \right) + \sum_{(i,j) \in \mathcal{G}} a_{ij} s_i
+ \sum_{(k,l) \in \mathcal{S}^* \cap \mathcal{S}^h} \sum_{(i,j) \in (S_k, T_l) \cap \mathcal{G}} c_{ij} + \sum_{(k,l) \in \mathcal{S}^* \cap \mathcal{S}^h} w_{kl}.
\]

**Proof.** Suppose the optimal routing ships items $(i,j) \in (S_k, T_l) \cap \tilde{D}^*$ for $(k,l) \in (B,B)$ on path $P^*_kl$. For each $(p,q) \in (B,B)$, let the set of items shipped on arc $\tilde{u}_{pq}$ by the optimal routing be

\[
Q^*_{pq} = \{(i,j) \in (S_k, T_l) \cap \tilde{D}^* \mid \tilde{u}_{pq} \in P^*_kl, (k,l) \in (B,B)\}.
\]
Thus, the cost of an optimal solution is

\[ z^* = \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S} \sum_{j \in D^*} a_{ij} s_i + \sum_{j \in T} \sum_{i \in D^*} a_{ij} t_j + \sum_{(p,q) \in (B,B)} \sum_{(i,j) \in Q^*_{pq}} a_{ij} u_{pq} \]

For any breakbulk pair \((p,q) \in (B,B)\), observe that arc \(\bar{u}_{pq}\) can not be used in the breakbulk path which begins with \(q\) or ends with \(p\). Consequently, the arc \(\bar{u}_{pq}\) can be included in at most \((b-1)(b-2)+1\) breakbulk paths. As a result, for \(b \geq 3\),

\[
(b-1)(b-2)+1 \geq 2 \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S} \sum_{j \in D^*} a_{ij} s_i + \sum_{j \in T} \sum_{i \in D^*} a_{ij} t_j + \sum_{(k,l) \in B^*} \sum_{(i,j) \in (S_k,T_l) \cap D^*} u_{pq} \geq 2 \sum_{(i,j) \in D^*} c_{ij} + \sum_{i \in S} \sum_{j \in D^*} a_{ij} s_i + \sum_{j \in T} \sum_{i \in D^*} a_{ij} t_j + \sum_{(k,l) \in B^*} \sum_{(i,j) \in (S_k,T_l) \cap D^*} a_{ij} u_{kl} \geq 2 \sum_{(i,j) \in D^*} c_{ij} + \sum_{j \in T^*} t_j + \sum_{i \in S^*} \sum_{i \in D^*} a_{ij} s_i + \sum_{i \in S^*} w_{kl} = 2 \sum_{(i,j) \in D^*} c_{ij} + \sum_{j \in T^*} t_j + \sum_{i \in S^*} a_{ij} s_i + \sum_{(i,j) \in B^*} w_{kl} = 2 \sum_{(i,j) \in D^*} c_{ij} + \sum_{j \in T^*} t_j + \sum_{i \in S^*} a_{ij} s_i + \sum_{(i,j) \in B^*} w_{kl}. \]

(6.3)

Similar to (6.2), the first term on the right hand side of (6.3),

\[
\sum_{(i,j) \in D^* \cap G} c_{ij} \geq \sum_{j \in T^* \cap k} \sum_{i \in G} c_{ij} + \sum_{j \in T^* \cap k} \left( t_j + \sum_{i \in G} a_{ij} (s_i + \bar{s}_{ij}) \right) + \sum_{j \in T^* \cap k} \sum_{i \in D^* \cap G} a_{ij} (s_i + \bar{s}_{ij}) \geq \sum_{j \in T^* \cap k} \sum_{i \in G} c_{ij} + \sum_{j \in T^* \cap k} \left( t_j + \sum_{i \in G} a_{ij} s_i \right) + \sum_{j \in T^* \cap k} \sum_{i \in D^* \cap G} a_{ij} s_i. \] (6.4)

Also, for each \((k,l) \in B^*\), all items \((i,j) \in (S_k,T_l)\) are shipped directly, which implies
that $D^* \cap \bar{G} \cap (S_k, T_l) = \bar{G} \cap (S_k, T_l)$. Hence,

$$\sum_{(i,j)\in D^* \cap \bar{G}} c_{ij} = \sum_{(k,l)\in \bar{B} \cap \bar{G}} \sum_{(i,j)\in (S_k, T_l) \cap \bar{G}} c_{ij} + \sum_{(k,l)\in \bar{B} \cap \bar{G}} \sum_{(i,j)\in (S_k, T_l) \cap \bar{G}} c_{ij} + \sum_{(k,l)\in \bar{B} \cap \bar{G}} \sum_{(i,j)\in (S_k, T_l) \cap \bar{G}} c_{ij}$$

$$\geq \sum_{(k,l)\in \bar{B} \cap \bar{G}} \sum_{(i,j)\in (S_k, T_l) \cap \bar{G}} c_{ij} + \sum_{(k,l)\in \bar{B} \cap \bar{G}} \sum_{(i,j)\in (S_k, T_l) \cap \bar{G}} c_{ij} \geq \sum_{(k,l)\in \bar{B} \cap \bar{G}} \sum_{(i,j)\in (S_k, T_l) \cap \bar{G}} c_{ij}. \quad (6.5)$$

Inequality (6.5) follows because $\bar{D} \subseteq \bar{G}$. For each $(k, l) \in (B, B)$, if $(S_k, T_l) \cap \bar{D} \neq \emptyset$, then some item $(i, j) \in (S_k, T_l)$ is shipped on arc $\bar{u}_{kl}$ in the heuristic solution. Thus, by Step 3 of HC,

$$\sum_{(k,l)\in \bar{B} \cap \bar{G}} \sum_{(i,j)\in (S_k, T_l) \cap \bar{G}} c_{ij} \geq \sum_{(k,l)\in \bar{B} \cap \bar{G}} \sum_{(i,j)\in (S_k, T_l) \cap \bar{G}} a_{ij}w_{kl} \geq \sum_{(k,l)\in \bar{B} \cap \bar{G}} w_{kl}.$$ 

Therefore, from (6.5),

$$\sum_{(i,j)\in D^* \cap \bar{G}} c_{ij} > \sum_{(k,l)\in \bar{B} \cap \bar{G}} \sum_{(i,j)\in (S_k, T_l) \cap \bar{G}} c_{ij} + \sum_{(k,l)\in \bar{B} \cap \bar{G}} w_{kl}. \quad (6.6)$$

Consequently, from (6.3), (6.4), and (6.6),

$$\left( (b - 1)(b - 2) + 1 \right) z^* > \sum_{i\in T^* \cap \bar{G}} c_{ij} + \sum_{i\in T^* \cap \bar{G}} \left( t_j + \sum_{i\in \bar{G}} a_{ij}s_i \right) + \sum_{i\in T^* \cap \bar{G}} a_{ij}s_i + \sum_{i\in T^* \cap \bar{G}} w_{kl}$$

$$+ \sum_{i\in T^* \cap \bar{G}} a_{ij}s_i + \sum_{i\in T^* \cap \bar{G}} a_{ij}s_i + \sum_{i\in T^* \cap \bar{G}} w_{kl} + \sum_{i\in T^* \cap \bar{G}} w_{kl}$$

$$= \sum_{i\in T^* \cap \bar{G}} c_{ij} + \sum_{i\in T^* \cap \bar{G}} \left( t_j + \sum_{i\in \bar{G}} a_{ij}s_i \right) + \sum_{i\in T^* \cap \bar{G}} a_{ij}s_i + \sum_{i\in T^* \cap \bar{G}} w_{kl}.$$

\[ \square \]
Lemma 6.3

\[ z^h \leq \sum_{(i,j) \in G} c_{ij} + \sum_{i \in S \times n^h} \sum_{j \in G} c_{ij} + \sum_{j \in T \times n^h} \sum_{i \in G} c_{ij} + \sum_{(k,l) \in B \times s^h} \sum_{(i,j) \in (S_\delta, T_\delta) \cap G} c_{ij} \\
+ \sum_{i \in S \times u^h} s_i + \sum_{j \in T \times u^h} t_j + \sum_{(k,l) \in B \times u^h} w_{kl} \\
+ \sum_{i \in S \times u^h} \sum_{j \in G} a_{ij}(t_j + \hat{v}_{ij}) + \sum_{j \in T \times u^h} \sum_{i \in G} a_{ij}s_i. \]

Proof. Similar to equation (5.16),

\[ z^h \leq \sum_{(i,j) \in G} c_{ij} + \sum_{i \in S \times n^h} \sum_{j \in G} c_{ij} + \sum_{j \in T \times n^h} \sum_{i \in G} c_{ij} + \sum_{(k,l) \in B \times n^h} \sum_{(i,j) \in (S_\delta, T_\delta) \cap G} c_{ij} \\
+ \sum_{i \in S \times u^h} s_i + \sum_{j \in T \times u^h} t_j + \sum_{(k,l) \in B \times u^h} w_{kl} \\
+ \sum_{(i,j) \in G} a_{ij}(s_i + v_{ij} + t_j) - \sum_{i \in S \times n^h} \sum_{j \in G} a_{ij}(t_j + v_{ij}) - \sum_{j \in T \times n^h} \sum_{i \in G} a_{ij}s_i \\
\leq \sum_{(i,j) \in G} c_{ij} + \sum_{i \in S \times n^h} \sum_{j \in G} c_{ij} + \sum_{j \in T \times n^h} \sum_{i \in G} c_{ij} + \sum_{(k,l) \in B \times n^h} \sum_{(i,j) \in (S_\delta, T_\delta) \cap G} c_{ij} \\
+ \sum_{i \in S \times u^h} s_i + \sum_{j \in T \times u^h} t_j + \sum_{(k,l) \in B \times u^h} w_{kl} \\
+ \sum_{i \in S \times u^h} \sum_{j \in G} a_{ij}(t_j + \hat{v}_{ij}) + \sum_{j \in T \times u^h} \sum_{i \in G} a_{ij}s_i. \]

\[ \square \]

Theorem 6.1 The relative error of the solution generated by HM is \((b-1)(b-2)+1\).

Proof. Combining Lemma 6.1 and 6.2, we obtain

\[ \left( (b-1)(b-2)+2 \right) z^* > \sum_{i \in S \times n^h} \sum_{j \in G} c_{ij} + \sum_{i \in S \times u^h} \left( s_i + \sum_{j \in G} a_{ij}(t_j + \hat{v}_{ij}) \right) \\
+ \sum_{j \in T \times n^h} \sum_{i \in G} c_{ij} + \sum_{j \in T \times u^h} \left( t_j + \sum_{i \in G} a_{ij}s_i \right) \\
+ \sum_{(k,l) \in B \times n^h} \sum_{(i,j) \in (S_\delta, T_\delta) \cap G} c_{ij} + \sum_{(k,l) \in B \times u^h} w_{kl} \\
+ \sum_{(i,j) \in G} a_{ij}(s_i + \hat{v}_{ij} + t_j). \]

70
Since $c_{ij} \leq s_i + \hat{b}_{ij} + t_j$ for $(i, j) \in G$,

$$\sum_{(i,j) \in G} c_{ij} \leq \sum_{(i,j) \in G} a_{ij}(s_i + \hat{b}_{ij} + t_j).$$

Therefore, from Lemma 6.3, $((b - 1)(b - 2) + 2)z^* > z^H$. This implies that

$$\frac{z^H - z^*}{z^*} \leq (b - 1)(b - 2) + 1.$$ 

By Example 6.1, the bound is tight. □

6.3 Conclusion

In this chapter, we develop a heuristic procedure, HM, for the general multiple breakbulk problem. The heuristic finds the shortest path for each breakbulk pair, converts the problem to a clique breakbulk problem, and solves the problem with the heuristic developed in Chapter 5. The relative error of the solution generated by HM is bounded by $(b - 1)(b - 2) + 1$ and the bound is tight. An empirical study of HM given in Appendix C shows that HM performs well in practice.
CHAPTER 7

SUMMARY AND FUTURE WORK

A primary purpose of this research is to understand some basic structures of LTL freight delivery problems with breakbulk terminals. We study and provide polynomial time procedures for various systems with single or multiple breakbulk terminals. The problems are shown unary $\mathcal{NP}$-hard except for the single origin single breakbulk problem which is binary $\mathcal{NP}$-hard. The heuristics proposed in this research are simple and easy to implement.

There are several generalizations of our freight delivery problems that should be considered in future research. One important aspect to be considered is the time that items spent on the network. The time required for transporting an item from origin to destination constitutes customer’s satisfaction level. Many LTL carriers provide guaranteed time delivery services. The goal is to minimize the total shipping costs while satisfying service time constraints. Other important factors which might be considered include: handling cost at breakbulk terminals, non-uniform fleet (variable vehicle sizes), and vehicle balancing (returning empty trucks).
APPENDIX A

COMPARISON OF H1 AND H(G) WITH $G_j = A_j + \Delta$

We conduct an empirical study to show that, for any constant $\Delta$, if $g_j = a_j + \Delta$ for all $j \in N$, then $z^1 \leq z^g$ for $n \geq 3$. Suppose that the $n$ destination terminals in $R$ are randomly located in a unit square around the breakbulk terminal $b$. The items are shipped from the origin to each destination with items sizes from i.i.d. $\text{Uniform}(0,1)$. Let the cost $c_{sb} = 1$, and $c_{aj}$ and $c_{bj}$ for $j \in N$ be the Euclidean distance between the terminals.

A program written in Fortran is developed, which randomly generates instances of $R$. For each instance of $R$, an item is generated for each destination with size $a_j$ from $\text{Uniform}(0,1)$. Suppose the coordinates of breakbulk $b$ is $(0,0)$, two random numbers, $x$ from $\text{Uniform}(-0.5, 0.5)$ and $y$ from $\text{Uniform}(-0.5, 0.5)$, are generated for the coordinates of each destination. Thus, the distance between $s$ and $j$ and between $b$ and $j$ are

$$c_{aj} = \sqrt{x^2 + y^2} \quad \text{and} \quad c_{bj} = \sqrt{(x + 1)^2 + y^2}.$$

The cost difference between $H_1$ and $H(g)$, $z^1 - z^g$ is computed using the formula obtained in Lemma 3.2. Then, a hypothesis test is performed based on the null and alternative hypothesis

$$H_0 : z^1 - z^g \leq 0 \quad \text{and} \quad H_a : z^1 - z^g > 0.$$
We run the program for 10,000 replications for each combination of \( \Delta \) and \( n \). The results, \( t = \frac{\bar{x}_n - \mu}{\sigma / \sqrt{n}} \), are given in Table A.1. The number in parentheses indicates the Type-I error \( \alpha \)-level at which we reject \( H_0 \). No number is shown if we cannot reject \( H_0 \) with a significant level \( \alpha \leq .2 \).

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \Delta = -.10 )</th>
<th>( \Delta = -.05 )</th>
<th>( \Delta = -.01 )</th>
<th>( \Delta = .01 )</th>
<th>( \Delta = .05 )</th>
<th>( \Delta = .10 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>-6.864</td>
<td>-2.428</td>
<td>.651</td>
<td>2.027 (.05)</td>
<td>-2.071</td>
<td>-5.132</td>
</tr>
<tr>
<td>4</td>
<td>-5.473</td>
<td>-2.852</td>
<td>.159</td>
<td>-1.078</td>
<td>-3.418</td>
<td>-8.809</td>
</tr>
<tr>
<td>5</td>
<td>-6.883</td>
<td>-4.215</td>
<td>1.424 (.10)</td>
<td>1.257 (.20)</td>
<td>-4.171</td>
<td>-8.896</td>
</tr>
<tr>
<td>6</td>
<td>-10.501</td>
<td>-4.337</td>
<td>-1.603</td>
<td>-1.147</td>
<td>-4.996</td>
<td>-11.279</td>
</tr>
<tr>
<td>7</td>
<td>-10.601</td>
<td>-4.307</td>
<td>.881 (.20)</td>
<td>.549</td>
<td>-2.613</td>
<td>-9.896</td>
</tr>
<tr>
<td>8</td>
<td>-11.452</td>
<td>-4.242</td>
<td>1.085 (.20)</td>
<td>-1.848</td>
<td>-4.878</td>
<td>-12.859</td>
</tr>
<tr>
<td>9</td>
<td>-11.670</td>
<td>-5.257</td>
<td>-7.49</td>
<td>-1.61</td>
<td>-5.660</td>
<td>-14.010</td>
</tr>
<tr>
<td>10</td>
<td>-11.121</td>
<td>-4.552</td>
<td>-1.910</td>
<td>1.331 (.10)</td>
<td>-3.498</td>
<td>-14.442</td>
</tr>
<tr>
<td>11</td>
<td>-12.539</td>
<td>-4.283</td>
<td>-1.402</td>
<td>.380</td>
<td>-4.969</td>
<td>-14.806</td>
</tr>
<tr>
<td>12</td>
<td>-14.082</td>
<td>-6.495</td>
<td>-.261</td>
<td>-.866</td>
<td>-6.351</td>
<td>-17.458</td>
</tr>
<tr>
<td>13</td>
<td>-14.200</td>
<td>-6.397</td>
<td>-.278</td>
<td>1.041 (.20)</td>
<td>-6.508</td>
<td>-17.459</td>
</tr>
<tr>
<td>16</td>
<td>-17.975</td>
<td>-7.941</td>
<td>-2.081</td>
<td>-1.641</td>
<td>-5.275</td>
<td>-21.158</td>
</tr>
<tr>
<td>19</td>
<td>-21.712</td>
<td>-5.327</td>
<td>1.106 (.20)</td>
<td>1.022 (.20)</td>
<td>-7.543</td>
<td>-23.600</td>
</tr>
</tbody>
</table>

Table A.1: t-values of \( z^1 - z^\alpha \) with \( g_j = a_j + \Delta \) for 10,000 replications.

We find that \( H_0 \) can only be rejected when \( |\Delta| = .01 \). Since the t-value decreases as \( |\Delta| \) increases or as \( n \) increases, we conclude that \( z^1 < z^\alpha \) when \( |\Delta| \geq .05 \) with \( n \geq 3 \). However, for \( |\Delta| = .01 \), \( H_0 \) is rejected in some cases. When \( |\Delta| \) is very small, it is difficult to show that \( z^1 - z^\alpha \leq 0 \) because \( z^1 - z^\alpha \to 0 \) as \( |\Delta| \to 0 \). However, when we increase the number of replications from 10,000 to 100,000 for \( |\Delta| = .01 \), \( H_0 \)
is rejected by only one case at $\alpha = .1$. This confirms that $z^1 < z^g$ for any constant $\Delta \neq 0$ when $g_j = a_j + \Delta$ for all $j$. The results for $|\Delta| = .01$ with 100,000 replications are given in Table A.2.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$\Delta = -.01$</th>
<th>$\Delta = .01$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>-1.888</td>
<td>.050</td>
</tr>
<tr>
<td>4</td>
<td>-.264</td>
<td>-1.397</td>
</tr>
<tr>
<td>5</td>
<td>-.851</td>
<td>-.747</td>
</tr>
<tr>
<td>6</td>
<td>-2.255</td>
<td>-.489</td>
</tr>
<tr>
<td>7</td>
<td>-1.835</td>
<td>-2.104</td>
</tr>
<tr>
<td>8</td>
<td>-.809</td>
<td>-.558</td>
</tr>
<tr>
<td>9</td>
<td>-1.321</td>
<td>-1.582</td>
</tr>
<tr>
<td>10</td>
<td>-.333</td>
<td>-.853</td>
</tr>
<tr>
<td>11</td>
<td>-.202</td>
<td>-.953</td>
</tr>
<tr>
<td>12</td>
<td>-2.806</td>
<td>-.959</td>
</tr>
<tr>
<td>13</td>
<td>-1.922</td>
<td>-.201</td>
</tr>
<tr>
<td>14</td>
<td>-.061</td>
<td>-1.881</td>
</tr>
<tr>
<td>15</td>
<td>-4.212</td>
<td>-1.232</td>
</tr>
<tr>
<td>16</td>
<td>-.320</td>
<td>-1.945</td>
</tr>
<tr>
<td>17</td>
<td>-2.193</td>
<td>-2.830</td>
</tr>
<tr>
<td>18</td>
<td>-2.411</td>
<td>-2.779</td>
</tr>
<tr>
<td>19</td>
<td>1.324 (.1)</td>
<td>-1.663</td>
</tr>
<tr>
<td>20</td>
<td>-1.143</td>
<td>-1.376</td>
</tr>
</tbody>
</table>

Table A.2: $t$-values of $z^1 - z^g$ with $g_j = a_j + \Delta$ for 100,000 replications.
The FORTRAN program used for this experiment is given in the following:

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* Author: Joseph Chao
* Date : 8/18/96
* Description: This program generates random single origin, single
  bulk problems with user specified number of destinations.
* The two costs, Z1 and Zg with gj = aj + Delta where Delta is a
  constant given by the user, are compared. A hypothesis test,
  H0: Z1-Zg<=0, is performed to detect whether Z1 and Zg are
  significantly different at various confidence levels. This
  program allows the user to specify a random number seed, a range
  of number of destinations, and the number of replications.
* Variables:
  * SumG1 = Summation of Aj in G1
  * SumNG1 = Summation of Aj in ~G1
  * SumG = Summation of Aj in G
  * SumNG = Summation of Aj in ~G
  * SumGG1 = Summation of Aj in G\G1 or G1\G depending on Delta
  * SumCC = Summation of Csj-Cbj in G\G1 or G1\G depending on Delta
  * Z1Zg = z1-zg
  * SumZ1Zg = sum of z1-zg
  * SumZ1Zg2 = sum of (z1-zg)^2
  * MeanZ1Zg = mean of z1-zg
  * VarZ1Zg = variance of z1-zg
  * T = T-value of z1-zg
  * nDest = Number of destination terminals
  * Alpha = Confidence level of rejecting H0

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

PROGRAM Aj_d
PARAMETER(T80= 0.84, T90= 1.28, T95= 1.65, T99= 2.33)

CHARACTER INFILE*15, OUTFILE*15, Alpha*4
INTEGER nDestBgn, nDestEnd
REAL Aj, X, Y, Csj, Cbj, CC, Delta
REAL SumG1, SumG, SumGG1, SumNG, SumNG1, SumCC
REAL Z1Zg, SumZ1Zg, SumZ1Zg2, MeanZ1Zg, VarZ1Zg, T

* The input file should contain data in the following order:
  * iSeed, Delta, nDestBgn, nDestEnd, nRep
  * Delta, nDestBgn, nDestEnd, nRep ...
  *

WRITE(*,2)
2 FORMAT('Enter input file name: ',$)
READ(*,4)INFILE
WRITE(*,3)
3 FORMAT('Enter output file name: ',$)

76
READ(*,4)OUTFILE
4 FORMAT(A15)

OPEN(UNIT=1, FILE=INFILE, STATUS='OLD')
OPEN(UNIT=2, FILE=OUTFILE, STATUS='UNKNOWN')

READ(1,*)iSeed

*DO 9999 WHILE(.NOT. EOF(1))
*  Repeat the experiment for different data in the input file
*  until the end of file is reached.
*
READ(1,*) Delta, nDestBgn, nDestEnd, nRep

WRITE(*,102) iSeed, nRep, Delta
WRITE(2,102) iSeed, nRep, Delta
102 FORMAT('iSeed=',I11,4X,'nRep=',I6,4X,'Delta=',F4.2)
WRITE(2,*)'-----------------------------'

DO 1111 nDest = nDestBgn, nDestEnd
   SumZ1Zg = 0
   SumZ1Zg2= 0

DO 1222 IREP = 1, nRep
   SumG1 = 0.
   SumG = 0.
   SumGG1= 0.
   SumNG1= 0.
   SumNG = 0.
   SumCC = 0.

* Generate an instance of R with "nDest" destinations.
*
   IF(Delta.GE.0) THEN
      DO 1333 J = 1, nDest
         Aj = RAN(iSeed)
         X = RAN(iSeed)-.5
         Y = RAN(iSeed)-.5
         Cbj = SQRT(X*X+Y*Y)
         Csj = SQRT((X+1)**2+Y**2)
         CC = Csj - Cbj
         IF(CC .LE. Aj) THEN
            SumG1 = SumG1 + Aj
         ELSE IF(CC .LE. Aj+Delta) THEN
            SumGG1 = SumGG1 + Aj
            SumCC = SumCC + CC
         ELSE

77
SumNG = SumNG + Aj
ENDIF

1333 CONTINUE
SumNG1 = SumGG1 + SumNG
ELSE
  DO 1444 J = 1, nDest
    Aj = RAN(iSeed)
    X = RAN(iSeed) -.5
    Y = RAN(iSeed) -.5
    Cobj = SQRT(X*X+Y*Y)
    Csj = SQRT((X+1)**2+Y**2)
    CC = Csj - Cobj
    IF(CC .LE. Aj+Delta) THEN
      SumG = SumG + Aj
    ELSE IF(CC .LE. Aj) THEN
      SumGG1 = SumGG1 + Aj
      SumCC = SumCC + CC
    ELSE
      SumNG1 = SumNG1 + Aj
    ENDIF
  1444 CONTINUE
SumNG = SumGG1 + SumNG1
ENDIF

* Compute z1-zg.
*

IF(Delta .GE. 0) THEN
  Z1Zg = Ceiling(SumNG1) - Ceiling(SumNG) - SumCC
ELSE
  Z1Zg = Ceiling(SumNG1) - Ceiling(SumNG) + SumCC
ENDIF

SumZ1Zg = SumZ1Zg + Z1Zg
SumZ1Zg2 = SumZ1Zg2 + Z1Zg*Z1Zg

1222 CONTINUE

MeanZ1Zg = SumZ1Zg/nRep
VarZ1Zg = (SumZ1Zg2-nRep*MeanZ1Zg*MeanZ1Zg)/(nRep-1.)
T = MeanZ1Zg/SQRT(VarZ1Zg/nRep)
IF(T .GE. T99)THEN
  Alpha = '.01 '
ELSEIF(T .GE. T95)THEN
  Alpha = '.05 '
ELSEIF(T .GE. T90)THEN
  Alpha = '.1 '
ELSEIF(T .GE. T80)THEN
  Alpha = '.2 '
ELSE

78
Alpha = 'NONE'
ENDIF
WRITE(*,104) nDest, MeanZ1Zg, VarZ1Zg, T, Alpha
WRITE(2,104) nDest, MeanZ1Zg, VarZ1Zg, T, Alpha
104 FORMAT(' nDest=',I3,2X,'Mean=',F6.3,2X,'Var=',F6.3,2X,
      'T=',F7.3,5X,'Reject H0 at ',A4)
1111 CONTINUE
9999 CONTINUE
END
APPENDIX B

EMPIRICAL STUDY OF H1, H2 AND THE OPTIMAL ROUTING

We conduct an empirical study to show the difference between \(z^1\), \(z^2\), \(z^*\) and \(z^{LP}\). Suppose that the \(n\) destination terminals in \(R\) are randomly located in a square of perimeter \(r\) around the breakbulk terminal \(B\). A program written in Fortran 90 is developed, which randomly generates instances of \(R\) using the same generation scheme as in Appendix A. A mixed integer programming package, LP_SOLVE, by Berkelaar(1996) is used to obtain an optimal solution, and a subroutine in IMSL mathematical library is used to obtain a linear relaxation solution for the problems. The costs, \(z^1\), \(z^2\), \(z^*\) and \(z^{LP}\), and their differences are computed.

For each combination of \(n = 5, 10, 50, 100, 500\) and \(r = .1, .3, .5, .9\), we obtain 100 data samples. The average cost differences, \(\mu_{12} = E(z^1 - z^2)\), \(\mu_{2*} = E(z^2 - z^*)\) and \(\mu_{2L} = E(z^2 - z^{LP})\), and the average relative differences, \(\rho_{12} = E(z^1 - z^2)/E(z^2)\), \(\rho_{2*} = E(z^2 - z^*)/E(z^*)\) and \(\rho_{2L} = E(z^2 - z^{LP})/E(z^{LP})\), are given in Table B.1.

We find that H2 performs very well on average. The relative error of H2, \(\mu_2^*\) is less than .23 in all cases for \(n \leq 50\). In the case of \(n > 50\), we do not have optimal solutions due to resource constraints. Since \(\mu_{2L} < .43\), the absolute error of H2 is no more than .43. For the relative error of H2, \(\rho_2^* < .033\) in all cases studied. Also, \(\rho_2^*\) goes to zero rapidly as \(n\) goes to infinity. The difference between H1 and H2 is less than .5. As a result, H1 also performs well on average, especially when \(n\) is large.
<table>
<thead>
<tr>
<th>r</th>
<th>n</th>
<th>$\mu_{12}$</th>
<th>$\mu_{2L}$</th>
<th>$\mu_{2*}$</th>
<th>$\rho_{12}$</th>
<th>$\rho_{2L}$</th>
<th>$\rho_{2*}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>0.00000</td>
<td>0.47270</td>
<td>0.08199</td>
<td>0.00000</td>
<td>0.17481</td>
<td>0.02424</td>
</tr>
<tr>
<td>.10</td>
<td>10</td>
<td>0.00000</td>
<td>0.46161</td>
<td>0.13613</td>
<td>0.00000</td>
<td>0.08239</td>
<td>0.02226</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.00000</td>
<td>0.42433</td>
<td>0.19173</td>
<td>0.00000</td>
<td>0.03775</td>
<td>0.01634</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.00881</td>
<td>0.35941</td>
<td>0.22654</td>
<td>0.00026</td>
<td>0.01282</td>
<td>0.00797</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.05029</td>
<td>0.42491</td>
<td>-</td>
<td>0.00088</td>
<td>0.00745</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>0.12558</td>
<td>0.37307</td>
<td>-</td>
<td>0.00044</td>
<td>0.00131</td>
<td>-</td>
</tr>
<tr>
<td>.30</td>
<td>5</td>
<td>0.00000</td>
<td>0.42733</td>
<td>0.11972</td>
<td>0.00000</td>
<td>0.13428</td>
<td>0.03268</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.05361</td>
<td>0.39034</td>
<td>0.15531</td>
<td>0.00766</td>
<td>0.05849</td>
<td>0.02253</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.15375</td>
<td>0.34120</td>
<td>0.16981</td>
<td>0.01075</td>
<td>0.02507</td>
<td>0.01217</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.24623</td>
<td>0.22167</td>
<td>0.13547</td>
<td>0.00722</td>
<td>0.00649</td>
<td>0.00396</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.24126</td>
<td>0.22590</td>
<td>-</td>
<td>0.00354</td>
<td>0.00331</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>0.37420</td>
<td>0.16634</td>
<td>-</td>
<td>0.00109</td>
<td>0.00048</td>
<td>-</td>
</tr>
<tr>
<td>.50</td>
<td>5</td>
<td>0.03249</td>
<td>0.40721</td>
<td>0.09713</td>
<td>0.00816</td>
<td>0.10601</td>
<td>0.02301</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.11667</td>
<td>0.32894</td>
<td>0.12827</td>
<td>0.01470</td>
<td>0.04254</td>
<td>0.01591</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.26825</td>
<td>0.24973</td>
<td>0.10420</td>
<td>0.01725</td>
<td>0.01621</td>
<td>0.00669</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.31518</td>
<td>0.18837</td>
<td>0.10044</td>
<td>0.00808</td>
<td>0.00486</td>
<td>0.00258</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.31629</td>
<td>0.12867</td>
<td>-</td>
<td>0.00405</td>
<td>0.00164</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>0.43646</td>
<td>0.06696</td>
<td>-</td>
<td>0.00112</td>
<td>0.00007</td>
<td>-</td>
</tr>
<tr>
<td>.90</td>
<td>5</td>
<td>0.03311</td>
<td>0.41740</td>
<td>0.12212</td>
<td>0.00765</td>
<td>0.09211</td>
<td>0.02500</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.13669</td>
<td>0.36105</td>
<td>0.11494</td>
<td>0.01384</td>
<td>0.03943</td>
<td>0.01228</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.21691</td>
<td>0.28674</td>
<td>0.11790</td>
<td>0.01150</td>
<td>0.01517</td>
<td>0.00607</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.21891</td>
<td>0.20815</td>
<td>0.10712</td>
<td>0.00470</td>
<td>0.00452</td>
<td>0.00232</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.34837</td>
<td>0.15535</td>
<td>-</td>
<td>0.00371</td>
<td>0.00166</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>0.41581</td>
<td>0.08742</td>
<td>-</td>
<td>0.00089</td>
<td>0.00019</td>
<td>-</td>
</tr>
</tbody>
</table>

Table B.1: Average differences between $z^1$, $z^2$, $z^*$ and $z^{LP}$.

The FORTRAN program used for this experiment is given in the following:

******************************************************************************************************************************************
*                                                                                                                                  
* Author: Joseph Chao                                                                                                              
* Date : 8/25/96                                                                                                                   
* Description: This program generates random single origin, single breakbulk problems with user specified number of destinations.  
* The costs, Z1, Z2, Z* and ZL, are computed. The LP solution is obtained using the subroutine DLPBS in IMSL, and the optimal solution is obtained using LP_SOLVE. However, the program bypass the optimal (IP) solution when n > 50 because of resource constraints. The average differences and relative
* differences between the costs are calculated.
* Subroutines:
* SVRG: IMSL routine which sorts a real array by algebraically
* increasing value and return the permutation that
* rearranges the array.
* DLPRS: IMSL routine which solves a linear programming problem via
* the revised simplex algorithm
* Variables:
* SumG1Cj = Summation of C_ij in G_1
* SumNG1Aj = Summation of A_j in \sim G_1
* SumNG1Cbj = Summation of C_bj in \sim G_1
* SumG2Cj = Summation of C_ij in G_2
* SumNG2Aj = Summation of A_j in \sim G_2
* SumNG2Cbj = Summation of C_bj in \sim G_2
* SumCC = Summation of C_ij-C_bj in G_2\setminus G_1
* Z1Z2 = z_1-z_2
* Z1ZL = z_1-z_1p
* Z2ZL = z_2-z_1p
* Z1ZI = z_1-z^*
* Z2ZI = z_2-z^*
* Sum12 = sum of z_1-z_2
* Sum1L = sum of z_1-z_1p
* Sum2L = sum of z_2-z_1p
* Sum1I = sum of z_1-z^*
* Sum2I = sum of z_2-z^*
* Mean12 = mean of z_1-z_2
* Mean1L = mean of z_1-z_1p
* Mean2L = mean of z_2-z_1p
* Mean1I = mean of z_1-z^*
* Mean2I = mean of z_2-z^*
* Rel12 = (z_1-z_2)/z_2
* Rel1L = (z_1-z_1p)/z_1p
* Rel2L = (z_2-z_1p)/z_1p
* Rel1I = (z_1-z^*)/z^*
* Rel2I = (z_2-z^*)/z^*
* MeanRel12 = mean of (z_1-z_2)/z_2
* MeanRel1L = mean of (z_1-z_1p)/z_1p
* MeanRel2L = mean of (z_2-z_1p)/z_1p
* MeanRel1I = mean of (z_1-z^*)/z^*
* MeanRel2I = mean of (z_2-z^*)/z^*
* nDest = Number of destination terminals

**********************************************************************************************

PROGRAM h1h2lpip

USE MSIMSL
USE PORTLIB

PARAMETER (n=500)

82
CHARACTER INFIL*15, OUTFILE*15
CHARACTER lpsolvefile*58, solnfile*58, dirname*50
CHARACTER command*200, valuestr*100, lpsoln*20
INTEGER posn
INTEGER nDest, NumG1
REAL A(n), Cs(n), Cb(n), CC(n), CRatio(n), SRes(n)
INTEGER G1(n)
REAL X, Y, R, SumCsj, SumCbj, SumAj, Space
REAL SumNG1Aj, SumG1Csj, SumNG1Cbj
REAL SumCC, Z1, Z2, ZL, ZI
REAL Z1Z2, Sum12, Mean12, SumRel12, MeanRel12
REAL Z1L, Sum1L, Mean1L, SumRel1L, MeanRel1L
REAL Z2L, Sum2L, Mean2L, SumRel2L, MeanRel2L
REAL Z1ZI, Sum1I, Mean1I, SumRel1I, MeanRel1I
REAL Z2ZI, Sum2I, Mean2I, SumRel2I, MeanRel2I

INTEGER M, LDA, IRTYPE(1)
REAL B(1), AMatrix(n+1), XLB(n+1), XUB(n+1), C(n+1)
REAL XSOL(n+1), DSOL(1)
PARAMETER (M=1, LDA=M)
DATA B/0./, IRTYPE/1/

* The input file should contain data in the following order:
* iSeed,
* R, nDest, nRep
* R, nDest, nRep ...

* WRITE(*,2)
  FORMAT(‘ Enter input file name: ’,$)
  READ(*,4)INFILE
  WRITE(*,3)
  FORMAT(‘ Enter output file name: ’,$)
  READ(*,4)OUTFILE
  FORMAT(A15)
  OPEN(UNIT=1, FILE=INFILE, STATUS=’OLD’)
  OPEN(UNIT=2, FILE=OUTFILE, STATUS=’UNKNOWN’)

  READ(1,*)iSeed
  WRITE(2,101) iSeed
  WRITE(*,11)
  WRITE(*,12)
  WRITE(2,11)
  WRITE(2,12)
  FORMAT(’ R   nDest   Mean12   MeanRel12’,
          ’   Mean1L   Rel1L   Mean2L   Rel2L’,
          ’   Mean1I   Rel1I   Mean2I   Rel2I’)
  FORMAT(’-----------------------------------’,
          ’-----------------------------------’,
          ’-----------------------------------’)

83
DO 9999 WHILE(.NOT. EOF(1))

* Repeat the experiment for different data in the input file
* until the end of file is reached.
*
READ(1,*) R, nDest, nRep

Sum12 = 0.
Sum1L = 0.
Sum2L = 0.
Sum1I = 0.
Sum2I = 0.
SumRel12 = 0.
SumRel1L = 0.
SumRel2L = 0.
SumRel1I = 0.
SumRel2I = 0.

DO 1020 IREP = 1, nRep
   SumG1-Csj = 0.
   SumNG1-Aj = 0.
   SumNG1-Cbj = 0.
   SumCC = 0.
   NumG1 = 0
   SumCsj = 0.
   SumCbj = 0.
   SumAj = 0.

   DO 1030 j = 1, nDest

* Generate an instance of R with "nDest" destinations.

   A(j) = RAN(iSeed)
   X = (RAN(iSeed) - .5) * 2.* R
   Y = (RAN(iSeed) - .5) * 2.* R
   Cb(j) = SQRT(X*X+Y*Y)
   Cs(j) = SQRT((X+1)**2+Y**2)

* Compute Z1.

   CC(j) = Cb(j) - Cs(j)
   IF(-CC(j) .GT. A(j)) THEN
   * Ship through the breakbulk.
     SumNG1-Aj = SumNG1-Aj + A(j)
     SumNG1-Cbj = SumNG1-Cbj + Cb(j)
   ELSE
   * Ship directly.
     SumG1-Csj = SumG1-Csj + Cs(j)
     NumG1 = NumG1 + 1
     G1(NumG1) = j
CRatio(NumG1) = CC(j)/A(j)
ENDIF

SumCsj = SumCsj + Cs(j)
SumCbj = SumCbj + Cb(j)
SumAj = SumAj + A(j)

1030 CONTINUE
Z1G = SumG1Csj + Ceiling(SumNG1Aj) + SumNG1Cbj
ZDir = SumCsj
ZBK = Ceiling(SumAj) + SumCbj
Z1 = MIN(ZDir, ZBK, Z1G)

* Compute LP solution.

NVAR = nDest +1
DO 1040 i=1,nDest
   AMatrix(i) = A(i)
   C(i) = CC(i)
   XLB(i)=0.
   XUB(i)=1.
1040 CONTINUE
AMatrix(NVAR) = -1.
C(NVAR) = 1.
XLB(NVAR)= 0.
XUB(NVAR)= nDest
CALL DLPRS(M, NVAR, AMatrix, LDA, B, B, C, IRTYPE,
-     XLB, XUB, OBJ, XSOL, DSOL)
ZL = SumCsj + OBJ

IF (nDest > 50) THEN
   ZI = ZL
goto 3000
ENDIF

* Compute IP Solution

dirname = 'x:\user\chao\lpdos\'

* WRITE problem definition for lp-solve. Format needed is
* max: -x1 + 2 x2;
* C1: 2x1 + x2 < 5;
* -4x1 + 4 x2 <5;
* int x2,x1;
* lpsolvefile = TRIM(dirname) // 'h1h2lpip.lp'

OPEN(UNIT=3, FILE=lpsolvefile, STATUS='UNKNOWN')

85
nVar = nDest + 1
WRITE (3, '(A4, \')') 'min:
DO 2000 i=1, nVar
  WRITE (3, '(1X, SP, F15.8, A1, SS, I4.4, \')') C(i), 'x', i
2000 CONTINUE
WRITE (3, '(A1)') ';
DO 2010 i=1, nVar
  WRITE (3, '(1X,SP,F15.8,A1,SS,I4.4,\')') AMatrix(i), 'x', i
2010 CONTINUE
WRITE (3, '(A6)') ' <= 0;
DO 2020 i=1, nDest
  WRITE (3, '(1X, A1, SS, I4.4, A6)') 'x', i, ' <= 1;'  
2020 CONTINUE
WRITE (3, '(1X, A1, SS, I4.4, A5, I5, A1)') 'x', i, ' <= ', nDest, ';
DO 2030 i=1, nVar
  WRITE (3, '(1X, A5, SS, I4.4, A1)') 'int x', i, ';
2030 CONTINUE
CLOSE(UNIT=3)

* Shell out command to execute lp-solve

  solnfile = TRIM(dirname) // 'h1h2lp.out'
  WRITE (lpsoin, '(SP, I10.9, SS)') FLOOR(OBJ)
  command = TRIM(dirname) // 'ip_solve.exe -b' // TRIM(lpsoin)
  // '<' // TRIM(lpsovelfile) // '>' // TRIM(solnfile)

  i = system(TRIM(command))
  IF (i .eq. -1) THEN
    i = ierrno()
    PRINT *, 'Error ', i
    CALL EXIT(1)
ENDIF

* Parse lp-solve output to obtain the objective value.

  OPEN(UNIT=4, FILE=solnfile, STATUS='OLD')
  READ(4,'(A100)') valuestr
  CLOSE(UNIT=4)
  posn = SCAN(valuestr, ':', Back = .True.) + 1
  valuestr = valuestr(posn:)
  READ (valuestr, *) ZI
  ZI = SumCsj + ZI

* Compute Z2, Z1-Z2, Z1-ZL, Z2-ZL, Z1-ZI and Z2-ZI.
3000  Z2 = Z1
       IF ((ABS(Z1-Z1G) .LE. .00001) .AND. (NumG1 .GT. 0)) THEN
*   Sort items in G1 according to the weighted cost saving
       CALL SVRGPS(NumG1, CRatio, sRes, G1)
       Space = CEILING(SumNG1Aj) - SumNG1Aj
       DD 1050 i = 1, NumG1
           IF (CC(G1(i)) .LT. 0 .AND. A(G1(i)) .LE. Space) THEN
*       Ship the direct item in H1 through BK
*       if the item fits in the space and Csj > Cbj.
           Z2 = Z2 + CC(G1(i))
           Space = Space - A(G1(i))
       ENDDIF
1050  CONTINUE
       ENDF

       Z1Z2 = Z1 - Z2
       Z1ZL = Z1 - ZL
       Z2ZL = Z2 - ZL
       Z1IZ = Z1 - ZI
       Z2ZI = Z2 - ZI

       Sum12 = Sum12 + Z1Z2
       Sum1L = Sum1L + Z1ZL
       Sum2L = Sum2L + Z2ZL
       Sum1I = Sum1I + Z1IZ
       Sum2I = Sum2I + Z2ZI

       SumRel12 = SumRel12 + Z1Z2 / Z2
       SumRel1L = SumRel1L + Z1ZL / ZL
       SumRel2L = SumRel2L + Z2ZL / ZL
       SumRel1I = SumRel1I + Z1IZ / ZI
       SumRel2I = SumRel2I + Z2ZI / ZI

1020  CONTINUE

       Mean12 = Sum12 / nRep
       Mean1L = Sum1L / nRep
       Mean2L = Sum2L / nRep
       Mean1I = Sum1I / nRep
       Mean2I = Sum2I / nRep

       MeanRel12 = SumRel12 / nRep
       MeanRel1L = SumRel1L / nRep
       MeanRel2L = SumRel2L / nRep
       MeanRel1I = SumRel1I / nRep
       MeanRel2I = SumRel2I / nRep

       WRITE(*,104) R, nDest, Mean12, MeanRel12,
       -       Mean1L, MeanRel1L, Mean2L, MeanRel2L,
       -       Mean1I, MeanRel1I, Mean2I, MeanRel2I

87
WRITE(2,104) R, nDest, Mean12, MeanRel12,
    Mean1L, MeanRel1L, Mean2L, MeanRel2L,
    Mean1I, MeanRel1I, Mean2I, MeanRel2I
104 FORMAT(‘ ’, F4.2, 2X, I5, 3X, F8.5, 3X, F8.5,
    3X, F8.5, 3X, F8.5, 3X, F8.5, 3X, F8.5,
    3X, F8.5, 3X, F8.5, 3X, F8.5, 3X, F8.5)

9999 CONTINUE

END

*****************************************************************************
APPENDIX C

EMPIRICAL STUDY FOR PROCEDURE HM

We conduct an empirical study for general breakbulk problems to demonstrate the performance of HM. Problems with four and six breakbulks are constructed. For each problem instance, the costs of the solution generated by procedure HM ($z^h$) and the LP relaxation solution ($z^{LP}$) are computed to obtain the relative error, $(z^h - z^{LP}) / z^{LP}$.

For the problems with four breakbulk terminals, one breakbulk is randomly located in each of the four square areas as shown in Figure C.1. There are 10 arcs shown in the figure between the breakbulk pairs. Figure C.1 also shows the problems with six breakbulk terminals, which has 14 arcs between the breakbulks. Each EOL terminal is randomly located in one of the square areas, and is assigned to the breakbulk that is located in the same square as the EOL terminal. The costs on the arcs between breakbulks and EOL terminals are calculated based on the Euclidean distance. The origin and destination terminals of each item are assigned randomly among all EOL terminals. Each item size is generated from Uniform(0,1).

FORTRAN programs for both the four and six breakbulk problems are developed, which randomly generate instances of $R_m$. As we formulate in Chapter 6, $R_m$ is a large-scaled integer programming problem. As an example, the formulation for the problem of 4 breakbulks, 30 EOL's and 200 items has 2270 decision variables and 870 constraints plus the integer constraints on the variables. For the problem with
Figure C.1: Networks for $R_m$ with four and six breakbulks.

6 breakbulks, 50 EOL’s and 600 items, we have 9114 decision variables and 3714 constraints. Because of the large size of $R_m$, LP solution is used as a lower bound for the solution generated by heuristic procedure HM. The linear programming package, LP_SOLVE (Berkelaar, 1996), is used to obtain a linear relaxation solution for each problem instance. We run the experiment of 4 breakbulk with 30 random EOL terminals. For each 200 and 500 random items, we generate and solve 10 problem instances. The average relative differences, $\rho = E(\z^h - \z^{LP})/E(\z^{LP})$, are given in Table C.1.

Table C.1 demonstrates that HM performs well on average. The relative error of HM is $\rho = 0.05208$ for the problem with 4 breakbulks, 30 EOL’s and 500 items. The study also shows that the relative error decreases as the number of items increases.
<table>
<thead>
<tr>
<th>No. of Breakbulks</th>
<th>No. of EOL's</th>
<th>No. of Items</th>
<th>$\rho$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>30</td>
<td>200</td>
<td>0.10511</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>500</td>
<td>0.05208</td>
</tr>
<tr>
<td>6</td>
<td>50</td>
<td>600</td>
<td>0.07866</td>
</tr>
</tbody>
</table>

Table C.1: Average relative differences of $x^h$ and $x^{LP}$.

The relative error for the problem with 6 breakbulks, 50 EOL's and 600 items is 0.07866. Note that 600 items is relative small for 50 EOL terminals which can have potentially 2,500 items. Due to computational resource constraints, we have not been able to perform a complete experiment. However, this study indicates that the performance of HM improves as the number of items increases.
The FORTRAN program used for the experiment with four breakbulks is given below. The program for the problems with six breakbulks is based on this program with minor modification.

***************************************************************
*  *  Author: Joseph Chao  *  *  Date : 9/10/96  
* Description: This program generates random four breakbulk problems
* with pre-defines breakbulk configuration as described in
* Appendix C. The location of each breakbulk, the costs on the arcs
* between breakbulks, and the cost of the shortest path for each
* breakbulk pair, are input from a file. The input file also
* dictates the number of EOL's and the number of items to be
* generated.
*  
***************************************************************
PROGRAM HMBK4LPS

USE PORTLIB

integer iSeed, nBK, nBkArc, maxEOL, maxItem, maxCnstr, maxVar
parameter(nBK=4, maxEOL=30, maxItem=500, nBkArc=10)
parameter(maxCnstr = nBk*maxItem+2*maxEOL+nBkArc + 2*maxItem)
parameter(maxVar = maxItem + 2*maxEOL + (maxItem+1)*nBkArc)

character infile*15, outfile*15
integer nEOL, nItem, nRep, iRep, iEx, i, j, k, m, iRow, iBk,
-      iShift, vShift, vCount, NG(maxItem)
integer PrimBK(maxEOL), Orig(maxItem), Dest(maxItem)
real tmp, NumNG, Dh, NDh, Zh, ZL, ZhZL, RelZhZL,
-      SumZhZL, SumRelZhZL, MeanZhZL, MeanRelZhZL,
real A(maxItem), XEOL(maxEOL), YEOL(maxEOL),
-      S(maxEOL), T(maxEOL), C(maxEOL,maxEOL),
real oSumNGCC(maxEOL), oSumNGA(maxEOL), oSumNGAWT(maxEOL),
-      dSumNGCC(maxEOL), dSumNGA(maxEOL), dSumNGAWS(maxEOL),
-      SumNGSA(maxEOL), SumNGTA(maxEOL)
real XB(nBk), YB(nBk), U(nBk,nBk), W(nBk,nBk),
-      bkSumNGC(nBk,nBk), bkSumNGA(nBk,nBk),
-      bkSumNGAST(nBk,nBk), SumNGBKA(nBk,nBk)
real UB(nBkArc), CC(maxVar), AMtrx(maxCnstr,maxVar)
real SumCij, OBJ
integer V(maxItem,nBk,nBk), TotFlowVar(0:maxItem)
integer nFlowVar, MA, nVar

character lpsfile*58, solnfile*58, dirname*50
character command*200, valuestr*100
integer posn
* Obtain input and output file names
* The input file should contain:
  * iSeed
  * XB(i), YB(i), i = 1, nBK
  * U(i,j), i = 1, nBK; j = 1, nBK
  * W(i,j), i = 1, nBK; j = 1, nBK
  * nEOL, nItem, nRep
  * nEOL, nItem, nRep
* ...

WRITE(*,2)
2 FORMAT(' Enter input file name: ',$)
READ(*,4) infile
WRITE(*,3)
3 FORMAT(' Enter output file name: ',$)
READ(*,4) outfile
4 FORMAT(A15)

OPEN(UNIT=1, FILE=infile, STATUS='OLD')
OPEN(UNIT=2, FILE=outfile, STATUS='UNKNOWN')

READ(1,*)iSeed
WRITE(2,101) iSeed, nBK, nBkArc
101 FORMAT(' iSeed=',I11, 3x, 'nBK=', I2, 3x, 'nBkArc=', I3)

WRITE(*,11)
WRITE(*,12)
WRITE(2,11)
WRITE(2,12)

11 FORMAT(' nRep nEOL nItem MeanZhZL MeanRelZhZL')
12 FORMAT(' ------------------------------------------')

* Input breakbulk locations.
  do i = 1, nBK
    read(i,*)XB(i), YB(i)
  end do

* Input the cost on the arc between each breakbulk pair.
  do 110 i = 1, nBK
  do 110 j = 1, nBK
    read(1,*)U(i,j)
  end do
110
* Input the cost of the shortest path for each breakbulk pair.

    do 120 i = 1, nBK
    do 120 j = 1, nBK
       read(1,*) W(i,j)
    end do

* Define the cost of each arc between each breakbulk pair.

    UB(1) = U(1,2)
    UB(2) = U(1,3)
    UB(3) = U(1,4)
    UB(4) = U(2,1)
    UB(5) = U(2,3)
    UB(6) = U(3,1)
    UB(7) = U(3,2)
    UB(8) = U(3,4)
    UB(9) = U(4,1)
    UB(10) = U(4,3)

    DO 9999 WHILE(.NOT. EOF(1))

* Repeat the experiment for different set of nEOL, nItem and nRep
* in the input file until the end of file is reached.
* read(1,*) nEOL, nItem, nRep
write(*,*) nEOL, nItem, nRep

    SumZhZL = 0
    SumRelZhZL = 0

    do 1000 iRep = 1, nRep

* Generate each EOL location, determine its primary breakbulk, and
* calculate the cost between the EOL and its primary breakbulk.

    do i = 1, nEOL
       XEOL(i) = (RAN(iSeed) -.5) * 20
       YEOL(i) = (RAN(iSeed) -.5) * 20
       if (XEOL(i).lt.0 .and. YEOL(i).ge.0) then
          PrimBK(i) = 1
          S(i) = sqrt((XEOL(i)-XB(1))**2 + (YEOL(i)-YB(1))**2)
          T(i) = S(i)
       elseif (XEOL(i).ge.0 .and. YEOL(i).ge.0) then
          PrimBK(i) = 2
          S(i) = sqrt((XEOL(i)-XB(2))**2 + (YEOL(i)-YB(2))**2)
          T(i) = S(i)
       elseif (XEOL(i).ge.0 .and. YEOL(i).lt.0) then
          PrimBK(i) = 3

94
S(i) = sqrt((XEOL(i)-XB(3))**2 + (YEOL(i)-YB(3))**2)
T(i) = S(i)
else
PrimBK(i) = 4
S(i) = sqrt((XEOL(i)-XB(4))**2 + (YEOL(i)-YB(4))**2)
T(i) = S(i)
endif
end do
*
Calculate the direct cost for each O-D pair.
do 130 i = 1, nEOL
do 130 j = 1, nEOL
   C(i,j) = sqrt((XEOL(i)-XEOL(j))**2 + (YEOL(i)-YEOL(j))**2)
continue
*
Determine the size of each item and its Orig and Dest terminals.
do 140 k = 1, nItem
   A(k) = RAN(iSeed)
   Orig(k) = ceiling(RAN(iSeed) * nEOL)
   Dest(k) = ceiling(RAN(iSeed) * nEOL)
   do while (PrimBK(Orig(k)) .EQ. PrimBK(Dest(k)))
*
* Re-assign origin and destination terminals for the item
* if they have the same primary breakbulk.
*
   Orig(k) = ceiling(RAN(iSeed) * nEOL)
   Dest(k) = ceiling(RAN(iSeed) * nEOL)
   do iEx = 1, k-1
      if(Orig(k).EQ.Orig(iEx) .AND. Dest(k).EQ.Dest(iEx)) then
         Dest(k) = Orig(k)
         exit
      endif
   end do
continue
140
NumNG = 0
Dh = 0
NDh = 0

do k = 1, nItem
   tmp=S(Orig(k))+T(Dest(k))+W(PrimBK(Orig(k)),PrimBK(Dest(k)))
*
Check the first condition in HC for shipping items direct.
if (C(Orig(k),Dest(k)) .LE. A(k) * tmp) then
   Dh = Dh + C(Orig(k),Dest(k))
else
   NumNG = NumNG + 1
NG(NumNG) = k
endif
end do

do i = 1, nEOL
oSumNGC(i) = 0
oSumNGA(i) = 0
oSumNGAWT(i) = 0
dSumNGC(i) = 0
dSumNGA(i) = 0
dSumNGAWS(i) = 0
SumNGSA(i) = 0
SumNGTA(i) = 0

do k = 1, NumNG
if (Orig(NG(k)) .EQ. i) then
  oSumNGC(i) = oSumNGC(i) + C(i,Dest(NG(k)))
oSumNGA(i) = oSumNGA(i) + A(NG(k))
tmp = W(PrimBK(i),PrimBK(Dest(NG(k)))) + T(Dest(NG(k)))
oSumNGAWT(i) = oSumNGAWT(i) + A(NG(k)) * tmp
endif
if (Dest(NG(k)) .EQ. i) then
dSumNGC(i) = dSumNGC(i) + C(Orig(NG(k)),i)
dSumNGA(i) = dSumNGA(i) + A(NG(k))
tmp = W(PrimBK(Orig(NG(k))),PrimBK(i)) + S(Orig(NG(k)))
dSumNGAWS(i) = dSumNGAWS(i) + A(NG(k)) * tmp
endif
end do
end do

do i = 1, nEOL
- oSumNGC(i) .LE. ceiling(oSumNGA(i))*S(i)+oSumNGAWT(i)then
  * Satisfy the second condition in HC --
  * ship each item having orig terminal i direct.
do k = 1, NumNG
  do while ((Orig(NG(k)) .EQ. i) .AND. (k .LE. NumNG))
    Dh = Dh + c(i,Dest(NG(k)))
    if (k .LE. NumNG-1) then
      * Remove this item from set NG.
      do m = k, NumNG-1
        NG(m) = NG(m+1)
      end do
    endif
  NumNG = NumNG - 1
  end do
end do
end if
end do
end do

do i = 1, nEOL
if(dSumNGC(i) .GT. 0 .AND.
    - dSumNGC(i) .LE. ceiling(dSumNGA(i))*T(i)+dSumNGAWS(i))then
    * Ship each item that has dest terminal i directly.
    do k = 1, NumNG
        do while ((Dest(NG(k)) .EQ. i) .AND. (k .LE. NumNG))
            Dh = Dh + c(Orig(NG(k)),i)
            if (k .LE. NumNG-1) then
                * Remove this item from set NG.
                do m = k, NumNG-1
                    NG(m) = NG(m+1)
                end do
            endif
            NumNG = NumNG - 1
        end do
    end do
end if
end do
end do

do 310 i = 1, nBK
do 310 j = 1, nBK
    bkSumNGC(i,j) = 0
    bkSumNGA(i,j) = 0
    bkSumNGAST(i,j) = 0
    SumNGbkA(i,j) = 0
    if (i .ne. j) then
        do k = 1, NumNG
            if (PrimBK(Orig(NG(k))) .EQ. i .AND.
                - PrimBK(Dest(NG(k))) .EQ. j) then
                bkSumNGC(i,j)=bkSumNGC(i,j)+c(Orig(NG(k)),Dest(NG(k)))
                bkSumNGA(i,j)=bkSumNGA(i,j)+A(NG(k))
                tmp = S(Orig(NG(k))) + T(Dest(NG(k)))
                bkSumNGAST(i,j) = bkSumNGAST(i,j) + A(NG(k)) * tmp
            endif
        end do
    endif
    310 continue
end do
end do

if( (i .ne. j) .AND. (bkSumNGC(i,j) .GT. 0) .AND.
    - (bkSumNGC(i,j) .LE.
    - ceiling(bkSumNGC(i,j))*W(i,j) + bkSumNGAST(i,j))) then
    * Satisfy the third condition in HC --
    * ship each item with primary breakbulk pair (i,j) directly.
    do k = 1, NumNG
        do while((PrimBK(Orig(NG(k))).eq.i .AND.
            - PrimBK(Dest(NG(k))).eq.j) .AND. (k .le. NumNG))
            Dh = Dh + c(Orig(NG(k)),Dest(NG(k)))
            if (k .LE. NumNG-1) then
                * Remove this item from set NG.
                do m = k, NumNG-1
                    NumNG = NumNG - 1
                end do
            endif
        end do
    end do
end if
NG(m) = NG(m+1)
end do
endif
NumNG = NumNG - 1
end do
end do
endif
continue

* Ship remaining items through breakbulks.

    do k = 1, NumNG
        SumNGSA(Orig(NG(k))) = SumNGSA(Orig(NG(k))) + A(NG(k))
        SumNGTA(Dest(NG(k))) = SumNGTA(Dest(NG(k))) + A(NG(k))
        i = PrimBK(Orig(NG(k)))
        j = PrimBK(Dest(NG(k)))
        SumNGbkA(i,j) = SumNGbkA(i,j) + A(NG(k))
    end do

    do i = 1, nEOL
        NDh = NDh + ceiling(SumNGSA(i))*S(i)+ceiling(SumNGTA(i))*T(i)
    end do

    do 400 i = 1, nBK
        NDh = NDh + ceiling(SumNGbkA(i,j)) * W(i,j)
    400 continue

* Heuristic cost equals to the total costs of direct and
* indirect items.
* Zh = Dh + NDh

* Solve LP solution using LP_SOLVE.
* Construct objective function.
* do k = 1, nItem
    cc(k) = -c(Orig(k),Dest(k))
end do

    do i = 1, nEOL
        cc(nItem+i) = S(i)
        cc(nItem+nEOL+i) = T(i)
    end do

    do m = 1, nBkArc
        cc(nItem+2*nEOL+m) = UB(m)
    end do
* Setup decision variables for each item on arcs between breakbulks.

\[
\text{do k = 1, nItem}
\]  
\[
n\text{FlowVar} = 6
\]
\[
\text{if } (\text{PrimBK(\text{orig}(k))}.eq.1 \text{ AND } \text{PrimBK(\text{Dest}(k))}.eq.2) \text{ then}
\]
\[
V(k,1,2) = 1
\]
\[
V(k,1,3) = 2
\]
\[
V(k,1,4) = 3
\]
\[
V(k,3,2) = 4
\]
\[
V(k,3,4) = 5
\]
\[
V(k,4,3) = 6
\]
\[
V(k,2,1) = 0
\]
\[
V(k,2,3) = 0
\]
\[
V(k,3,1) = 0
\]
\[
V(k,4,1) = 0
\]
\[
\text{elseif } (\text{PrimBK(\text{orig}(k))}.eq.1 \text{ AND } \text{PrimBK(\text{Dest}(k))}.eq.3) \text{ then}
\]
\[
n\text{FlowVar} = 5
\]
\[
V(k,1,2) = 1
\]
\[
V(k,1,3) = 2
\]
\[
V(k,1,4) = 3
\]
\[
V(k,2,3) = 4
\]
\[
V(k,4,3) = 5
\]
\[
V(k,2,1) = 0
\]
\[
V(k,3,1) = 0
\]
\[
V(k,3,2) = 0
\]
\[
V(k,4,1) = 0
\]
\[
\text{elseif } (\text{PrimBK(\text{orig}(k))}.eq.1 \text{ AND } \text{PrimBK(\text{Dest}(k))}.eq.4) \text{ then}
\]
\[
V(k,1,2) = 1
\]
\[
V(k,1,3) = 2
\]
\[
V(k,1,4) = 3
\]
\[
V(k,2,3) = 4
\]
\[
V(k,3,2) = 5
\]
\[
V(k,3,4) = 6
\]
\[
V(k,2,1) = 0
\]
\[
V(k,3,1) = 0
\]
\[
V(k,4,1) = 0
\]
\[
V(k,4,3) = 0
\]
\[
\text{elseif } (\text{PrimBK(\text{orig}(k))}.eq.2 \text{ AND } \text{PrimBK(\text{Dest}(k))}.eq.1) \text{ then}
\]
\[
V(k,2,1) = 1
\]
\[
V(k,2,3) = 2
\]
\[
V(k,3,1) = 3
\]
\[
V(k,3,4) = 4
\]
\[
V(k,4,1) = 5
\]
\[
V(k,4,3) = 6
\]
\[
V(k,1,2) = 0
\]
\[
V(k,1,3) = 0
\]
\[
V(k,1,4) = 0
\]
\[
V(k,3,2) = 0
\]
\[
\text{elseif } (\text{PrimBK(\text{orig}(k))}.eq.2 \text{ AND } \text{PrimBK(\text{Dest}(k))}.eq.3) \text{ then}
\]
\begin{align*}
V(k,1,3) &= 1 \\
V(k,1,4) &= 2 \\
V(k,2,1) &= 3 \\
V(k,2,3) &= 4 \\
V(k,4,1) &= 5 \\
V(k,4,3) &= 6 \\
V(k,1,2) &= 0 \\
V(k,3,1) &= 0 \\
V(k,3,2) &= 0 \\
V(k,3,4) &= 0 \\
\text{elseif} \ (\text{PrimBK}(\text{orig}(k)).\text{eq.2} \ \&\ \text{PrimBK}(\text{Dest}(k)).\text{eq.4}) \ \text{then} \\
&\quad V(k,1,3) = 1 \\
&\quad V(k,1,4) = 2 \\
&\quad V(k,2,1) = 3 \\
&\quad V(k,2,3) = 4 \\
&\quad V(k,3,1) = 5 \\
&\quad V(k,3,4) = 6 \\
&\quad V(k,1,2) = 0 \\
&\quad V(k,3,2) = 0 \\
&\quad V(k,4,1) = 0 \\
&\quad V(k,4,3) = 0 \\
\text{elseif} \ (\text{PrimBK}(\text{orig}(k)).\text{eq.3} \ \&\ \text{PrimBK}(\text{Dest}(k)).\text{eq.1}) \ \text{then} \\
&\quad nFlowVar = 5 \\
&\quad V(k,2,1) = 1 \\
&\quad V(k,3,1) = 2 \\
&\quad V(k,3,2) = 3 \\
&\quad V(k,3,4) = 4 \\
&\quad V(k,4,1) = 5 \\
&\quad V(k,1,2) = 0 \\
&\quad V(k,1,3) = 0 \\
&\quad V(k,1,4) = 0 \\
&\quad V(k,2,3) = 0 \\
&\quad V(k,4,3) = 0 \\
\text{elseif} \ (\text{PrimBK}(\text{orig}(k)).\text{eq.3} \ \&\ \text{PrimBK}(\text{Dest}(k)).\text{eq.2}) \ \text{then} \\
&\quad V(k,1,2) = 1 \\
&\quad V(k,1,4) = 2 \\
&\quad V(k,3,1) = 3 \\
&\quad V(k,3,2) = 4 \\
&\quad V(k,3,4) = 5 \\
&\quad V(k,4,1) = 6 \\
&\quad V(k,1,3) = 0 \\
&\quad V(k,2,1) = 0 \\
&\quad V(k,2,3) = 0 \\
&\quad V(k,4,3) = 0 \\
\text{elseif} \ (\text{PrimBK}(\text{orig}(k)).\text{eq.3} \ \&\ \text{PrimBK}(\text{Dest}(k)).\text{eq.4}) \ \text{then} \\
&\quad V(k,1,2) = 1 \\
&\quad V(k,1,4) = 2 \\
&\quad V(k,2,1) = 3 \\
&\quad V(k,3,1) = 4 \\
&\quad V(k,3,2) = 5 \\
&\quad V(k,3,4) = 6
\end{align*}
V(k,1,3) = 0
V(k,2,3) = 0
V(k,4,1) = 0
V(k,4,3) = 0
elseif (PrimBK(orig(k)).eq.4 .AND. PrimBK(Dest(k)).eq.1) then
  V(k,2,1) = 1
  V(k,2,3) = 2
  V(k,3,1) = 3
  V(k,3,2) = 4
  V(k,4,1) = 5
  V(k,4,3) = 6
  V(k,1,2) = 0
  V(k,1,3) = 0
  V(k,1,4) = 0
  V(k,3,4) = 0
elseif (PrimBK(orig(k)).eq.4 .AND. PrimBK(Dest(k)).eq.2) then
  V(k,1,2) = 1
  V(k,1,3) = 2
  V(k,3,1) = 3
  V(k,3,2) = 4
  V(k,4,1) = 5
  V(k,4,3) = 6
  V(k,1,4) = 0
  V(k,2,1) = 0
  V(k,2,3) = 0
  V(k,3,4) = 0
elseif (PrimBK(orig(k)).eq.4 .AND. PrimBK(Dest(k)).eq.3) then
  V(k,1,2) = 1
  V(k,1,3) = 2
  V(k,2,1) = 3
  V(k,2,3) = 4
  V(k,4,1) = 5
  V(k,4,3) = 6
  V(k,1,4) = 0
  V(k,3,1) = 0
  V(k,3,2) = 0
  V(k,3,4) = 0
endif

TotFlowVar(0) = 0
TotFlowVar(k) = TotFlowVar(k-1) + nFlowVar
end do

* Calculate the numbers of decision variables and constraints.
* Reset the A matrix for each replication.

nVar = nItem + 2*nEOL + nBkArc + TotFlowVar(nItem)
MA = nBk*nItem + 2*nEOL + nBkArc + 2*nItem
do j = 1, nVar
  do i = 1, MA
    AMtrx(i,j) = 0
Construct flow conservation constraints for each item
* at each breakbulk terminal (nBk*nItem constraints).

vShift = nItem + 2*nEUL + nBkArc

do k = 1, nItem

* flow-in = flow-out at breakbulk 1

iRow = 4*(k-1)+1
if (PrimBK(orig(k)) .EQ. 1) then
   AMtrx(iRow,k) = 1
else
   if (V(k,2,1)>0) then
      AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,2,1)) = 1
   endif
   if (V(k,3,1)>0) then
      AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,3,1)) = 1
   endif
   if (V(k,4,1)>0) then
      AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,4,1)) = 1
   endif
endif

if (PrimBK(Dest(k)) .EQ. 1) then
   AMtrx(iRow,k) = -1
else
   if (V(k,1,2)>0) then
      AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,1,2)) = -1
   endif
   if (V(k,1,3)>0) then
      AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,1,3)) = -1
   endif
   if (V(k,1,4)>0) then
      AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,1,4)) = -1
   endif
endif

* flow-in = flow-out at breakbulk 2

iRow = 4*(k-1)+2
if (PrimBK(orig(k)) .EQ. 2) then
   AMtrx(iRow,k) = 1
else
   if (V(k,1,2)>0) then
      AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,1,2)) = 1
   endif
   if (V(k,3,2)>0) then
      AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,3,2)) = 1
   endif
endif
AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,3,2)) = 1
endif
endif

if (PrimBK(Dest(k)).EQ. 2) then
  AMtrx(iRow,k) = -1
else
  if (V(k,2,1)>0) then
    AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,2,1)) = -1
  endif
  if (V(k,2,3)>0) then
    AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,2,3)) = -1
  endif
endif

* flow-in = flow-out at breakbulk 3

iRow = 4*(k-1)+3
if (PrimBK(orig(k)).EQ. 3) then
  AMtrx(iRow,k) = 1
else
  if (V(k,1,3)>0) then
    AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,1,3)) = 1
  endif
  if (V(k,2,3)>0) then
    AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,2,3)) = 1
  endif
  if (V(k,4,3)>0) then
    AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,4,3)) = 1
  endif
endif

if (PrimBK(Dest(k)).EQ. 3) then
  AMtrx(iRow,k) = -1
else
  if (V(k,3,1)>0) then
    AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,3,1)) = -1
  endif
  if (V(k,3,2)>0) then
    AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,3,2)) = -1
  endif
  if (V(k,3,4)>0) then
    AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,3,4)) = -1
  endif
endif

* flow-in = flow-out at breakbulk 4

iRow = 4*(k-1)+4
if (PrimBK(orig(k)).EQ. 4) then
  AMtrx(iRow,k) = 1
else
    if (V(k,1,4)>0) then
        AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,1,4)) = 1
    endif
    if (V(k,3,4)>0) then
        AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,3,4)) = 1
    endif
endif

if (PrimBK(Dest(k)) .EQ. 4) then
    AMtrx(iRow,k) = -1
else
    if (V(k,4,1)>0) then
        AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,4,1)) = -1
    endif
    if (V(k,4,3)>0) then
        AMtrx(iRow,vShift+TotFlowVar(k-1)+V(k,4,3)) = -1
    endif
endif
end do

* Construct capacity constraints for each arc between each EOL and its primary breakbulk (2*nEOL constraints).

iShift = nBk * nItem

do k = 1, nItem
    AMtrx(iShift+Orig(k),k) = A(k)
    AMtrx(iShift+Orig(k),nItem+Orig(k)) = -1
    AMtrx(iShift+nEOL+Dest(k),k) = A(k)
    AMtrx(iShift+nEOL+Dest(k),nItem+nEOL+Dest(k)) = -1
end do

* Construct capacity constraints for each breakbulk pair (nBkArc constraints).

iShift = iShift + 2*nEOL

do m = 1, nBkArc
    do k = 1, nItem
        select case (m)
        case (1)
            iBk = V(k,1,2)
        case (2)
            iBk = V(k,1,3)
        case (3)
            iBk = V(k,1,4)
        case (4)
            iBk = V(k,2,1)
        end select
        AMtrx(iShift,iBk) = X(m)
    end do
end do
case (5)
    iBk = V(k,2,3)
case (6)
    iBk = V(k,3,1)
case (7)
    iBk = V(k,3,2)
case (8)
    iBk = V(k,3,4)
case (9)
    iBk = V(k,4,1)
case (10)
    iBk = V(k,4,3)
end select

if (iBk.gt.0) then
    AMtr(x(iShift+m,vShift+TotFlowVar(k-1)+iBk) = A(k)
end if
end do

AMtr(x(iShift+m,nItem+2*nEOL+m) = -1
end do

* Add 2*nItem logic constraints to smaller the gap between IP
* and LP. (If an item is shipped through breakbulks, at least
* one trailer is needed on the arc between its origin and
* primary breakbulk, and on the arc between its destination
* and primary breakbulk.)

iShift = iShift + nBkArc
Do k = 1, nItem
    AMtr(x(iShift+k,k) = 1
    AMtr(x(iShift+k,nItem+orig(k)) = -1
    AMtr(x(iShift+nItem+k,k) = 1
    AMtr(x(iShift+nItem+k,nItem+nEOL+dest(k)) = -1
end do

* Prepares problem file for LP_SOLVE to read and obtain an LP
* solution. The resulting file produced by LP_SOLVE is then
* parsed, and the objective value is returned.
* However, the location of LP_SOLVE, and its input and output
* file name is hard-coded in the program.
*
dirname = 'd:\lp20w95\'
*
* Write problem definition for lp-solve. Format needed is
*
* max: -x1 + 2x2;
* C1:  2x1 + x2 <= 5;
*    -4x1 + 4x2 <= 5;
*    x1    <= 1;
* 
x2 <= 1;
*

lpsfile = TRIM(dirname) // 'hmBk4lps.lp'

OPEN(UNIT=3, FILE=lpsfile, STATUS='UNKNOWN')

* Write objective function to the problem file.

WRITE (3, '(A4, \\)') 'min:'
do i = 1, nVar
   if (CC(i) .NE. 0.) then
      WRITE (3,99) CC(i), 'x', i
   endif
end do
WRITE (3, '(A1)') ';

* Write constraints to the problem file.

do i = 1, MA
   vCount = 0
   do j = 1, nVar
      if (AMtrx(i,j) .NE. 0.) then
         vCount = vCount + 1
         WRITE (3,99) AMtrx(i,j), 'x', j
      endif
   end do
   if (vCount .GT. 0) then
      WRITE (3, '(A6)') ' <= 0;'
   endif
end do

format(1X, SP, F15.8, A1, SS, I4.4, \\)

* Write upper bound constraints.

do i = 1, nItem
   WRITE(3, '(1X, A1, SS, I4.4, A6)') 'x', i, ' <= 1;' 
end do

do i = nItem+1, nItem+2*nEOL+nBkArc
   WRITE(3, '(1X, A1, SS, I4.4, A5, I5, A1)') 'x', i, ' <= ', 
      nItem, ';
end do

do i = nItem+2*nEOL+nBkArc+1, nVar
   WRITE(3, '(1X, A1, SS, I4.4, A6)') 'x', i, ' <= 1;'
end do

CLOSE(UNIT=3)

* Shell out command to execute LP_SOLVE.
solnfile = TRIM(dirname) // 'hmBk4lps.out'
command = TRIM(dirname) // 'lp_solve.exe' //
- ' ' // TRIM(lpsfile) // ' ' // TRIM(solnfile)

i = system(TRIM(command))
IF (i .eq. -1) THEN
   i = ierrno()
   PRINT *, 'Error ', i
   CALL EXIT(1)
ENDIF

* Parse lp-solve output to obtain the objective value.

OPEN(UNIT=4, FILE=solnfile, STATUS='OLD')
READ(4,'(A100)') valuestr
CLOSE(UNIT=4)
posn = SCAN(valuestr, ':', Back = .True.) + 1
valuestr = valuestr(posn:)
READ (valuestr, *) OBJ

SumCij = 0
DO k = 1, nItem
   SumCij = SumCij + C(Orig(k), Dest(k))
END DO
ZL = SumCij + OBJ

* Compute the different and the relative difference between the
* heuristic and LP solutions.

ZhZL = Zh - ZL
RelZhZL = ZhZL / ZL

write(*,195)iRep, ZL, ZhZL, RelZhZL
write(2,195)iRep, ZL, ZhZL, RelZhZL
195 format(1x, 'iRep=', i3, 2x, 'ZL=', F12.5, 2x, 2x,
   - 'ZhZL=', F12.5, 'RelZhZL=', F12.5)

SumZhZL = SumZhZL + ZhZL
SumRelZhZL = SumRelZhZL + RelZhZL

1000 continue

MeanZhZL = SumZhZL / nRep
MeanRelZhZL = SumRelZhZL / nRep

write(*,199) nRep, nEOL, nItem, MeanZhZL, MeanRelZhZL
write(2,199) nRep, nEOL, nItem, MeanZhZL, MeanRelZhZL
199 format(1x, i4, 2x, i3, 2x, i4, 2x, F12.5, 2x, F12.5)

9999 continue
close(1)
close(2)

END

******************************************************************************
REFERENCES


