

A FAMILY OF DOMINANCE FILTERS FOR MULTIPLE CRITERIA DECISION MAKING: CHOOSING THE RIGHT FILTER FOR A DECISION SITUATION

DISSERTATION

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

By

Naresh Sundaram Iyer, B.E., M.S.

* * * * *

The Ohio State University

2001

Dissertation Committee:

B. Chandrasekaran, Adviser

John R. Josephson

Marc E. Posner

Approved by

Adviser

Department of Computer
and Information Science

UMI Number: 3035521



UMI Microform 3035521

Copyright 2002 by ProQuest Information and Learning Company.

All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

© Copyright by
Naresh Sundaram Iyer
2001

ABSTRACT

Multiple Criteria Decision Making (MCDM) problems involve the selection of “good” alternatives from a set of alternatives, each of which is evaluated along multiple, and potentially conflicting, criteria. The criteria are intended to reflect the dimensions of outcome that matter to the decision-maker (DM). The decision-making process should select alternatives which optimize the outcomes most desired by the DM. Decision Support Systems (DSSs) are aids that enhance the decision-making capabilities of the DM in various ways. The DM may be modeled as a holder of preferences of various kinds. Decision support, then, entails the elicitation of these preferences and their application to the set of alternatives at hand. Ideal DSSs, in this view, must allow for the natural, accurate, and complete expression of preferences by the DM and apply, or help the DM apply, these preferences. Another aspect of decision-making is the wide variability in what we might call decision situations. These situations are characterized by the differences in the degrees to which optimality is essential to the DM, the time-pressure under which the decision is being made, the degree of pruning desired, the presence of uncertainty in criteria values, etc. DSSs that provide situation-specific support are more valuable.

In this work, we focus on the Seeker-Filter-Viewer (S-F-V) family of architectures and on their applicability as decision support architectures. The generic version of

this architecture makes use of the Pareto Dominance Filter to eliminate suboptimal alternatives. We explore Tolerance-Based Dominance Filters (TBDFs), which are based on decision rules similar to the Dominance rule but contain criteria-specific tolerances in the rule clauses. We analyze the applicability of TBDFs to a class of decision situations, with and without uncertainty in criteria values, and in the presence of a number of user-needs and other problem characteristics.

The goal is to develop a framework for mapping decision situations to appropriate instantiations of the S-F-V architecture. We present such a framework for the Filters and decision situations we consider in the dissertation. By using such a framework, the S-F-V architecture can cater to a larger class of decision problems and DMs than the earlier versions.

Dedicated to my parents

ACKNOWLEDGMENTS

This work is a product of five years of dedication, hard-work, and devotion of mine but these alone would always have fallen short if not for the support, guidance, inspiration, and advice of a few others. Hence, this document will be incomplete without my mentioning their names and the intangibles they provided.

I am immeasurably indebted to B. Chandrasekaran, my adviser, for being a constant source of knowledge, insight, encouragement, support, and inspiration over the past 5 years. His intellectual contribution to this work, both directly and indirectly, has been invaluable to my progress with it. I wish to thank him for allowing me the discretion of walking into his office anytime I had a question, and for giving me the opportunity to pursue my research under his guidance. He kept me going in my work in more ways than I can mention. His contribution to the clarity, notations, and conciseness of this document is immense and I wish to thank him for the time and effort he spent refining my untrained technical writing skills; I am a better writer for it today.

John Josephson has constantly influenced this dissertation in many ways. His presence as a mentor, friend, and philosopher, provided me with a number of insights in my work. I wish to thank him for all the brainstorming sessions I shared with him; I am better thinker for it today.

I am also grateful to Prof. Marc Posner for his useful comments on my work and results related to it. I am indebted to him for refining my work related to the Bayesian filter. I wish to also thank Prof. H. N. Nagaraja from the Department of Statistics, for advising me on the confidence estimate computations for the Monte-Carlo runs. My special thanks go out to my Department for providing me with financial support to pursue my work, and to the faculty in fostering a growth-conducive environment.

Over the years, a number of friends and colleagues, have supported me by participating in discussions of pertinence to this work. It is hard to imagine that these discussions did not set some wheels in motion. I thank Mark Carroll, Pankaj Choudhary, and Manoj Kaplinghat for being indirect influences on my work. Thanks are also in accord to Prasad, Arjun, Deepak, and Ketan, for being great friends and supportive in so many ways. Finally, I thank my family without whom I will have fallen astoundingly short of pursuing something so ambitious.

A great part of my early research was supported by a grant from DARPA. I wish to acknowledge the support of the DARPA RaDEO program (DARPA Order Number: D594, ONR Grant Number: N00014-96-1-0701).

VITA

October 19, 1970 Born - Bombay, India

1992 B.E. Electrical Engineering, VJTI,
Bombay

1996 M.S. Computer Science, University of
South Florida, Tampa.

PUBLICATIONS

John R. Josephson, B. Chandrasekaran, Mark Carroll, Naresh S. Iyer, Bryon Wasacz, Giorgio Rizzoni, QingYuan Li, and David A. Erb. (1998) An Architecture for Exploring Large Design Spaces. Proceedings of AAAI, Madison, Wisconsin, pp. 143-150.

Donald Houser, Jonny Harianto, B. Chandrasekaran, John R. Josephson, and Naresh S. Iyer. (2000). A Multi-Variable Approach to determine the “Best” Gear Design. Proceedings of the DETC2000, 2000 ASME Power Transmission and Gearing Conference, Baltimore, Maryland, pp 1-9.

Naresh S. Iyer, Abraham Kandel, and Moti Schneider. (2000) Feature-based fuzzy classification for interpretation of mammograms. Fuzzy Sets and Systems 114, pp. 271-280.

Naresh S. Iyer, and John R. Josephson (2001). Multicriterially Best Explanations. Lecture Notes in Computer Science: Proceedings of the Fourth International Conference on Discovery Science, Washington D.C. Springer-Verlag.

FIELDS OF STUDY

Major Field: Computer and Information Science

Studies in:

Artificial Intelligence	B. Chandrasekaran and John R. Josephson
Cognitive Science	James T. Todd
Theory of Computation	Rephael Wenger

TABLE OF CONTENTS

ABSTRACT	ii
DEDICATION	iv
ACKNOWLEDGMENTS	v
VITA	vii
LIST OF FIGURES	xiii

CHAPTER	PAGE
1 INTRODUCTION	1
1.1 Decision-making and Decision Support Systems	1
1.2 Multiple Criteria Decision Making (MCDM) as the norm	2
1.3 Modeling the Decision Maker	4
1.4 Existing approaches for solving the discrete alternative MCDM problem - A Brief Survey	7
1.5 Decision Situations: The need for situation-specific decision support	13
1.6 A Decision Support Architecture for MCDM	13
1.7 The Seeker-Filter-Viewer Architecture	14
1.7.1 The Seeker	15
1.7.2 The Filter	15
1.7.3 The Viewer	17
1.8 Motivation, Goals, and Outline of the Dissertation	19
1.8.1 Goals of the Dissertation	21
1.8.2 Outline of the Dissertation	21

2	PERFORMANCE ANALYSIS OF THE DOMINANCE FILTER	23
2.1	Introduction	23
2.2	Performance of the Dominance Filter	23
2.2.1	Filtering Efficiency of the Dominance Filter With Uncorrelated Criteria	24
2.2.2	Computational Time Complexity assuming uncorrelated criteria	27
2.3	Impact of correlated criteria	28
2.3.1	The Correlation Coefficient	29
2.3.2	Filtering Efficiency and correlation coefficient	30
2.4	Summary	34
3	DECISION SITUATIONS AND CHOICE FILTERS	36
3.1	Decision Situations	36
3.2	Choice Filters	37
3.2.1	Terminology and Notations	38
3.3	Tolerance-based Dominance Filters (TBDFs)	39
3.3.1	Single Pass Strict dominance tolerance filter- $ST(\epsilon)$	39
3.3.2	Twopass Strict dominance tolerance filter- $ST2(\epsilon)$	42
3.3.3	Multipass Strict dominance tolerance filter- $STm(\epsilon)$	44
3.3.4	Strongly-Strict dominance filter - $Sts(\epsilon)$	46
3.3.5	Superstrict dominance tolerance filter- $SS(\epsilon)$	49
3.3.6	DD-dominance tolerance filter- $DD(\epsilon)$	51
3.3.7	ϵ -Box Twopass Filter (ebtp-filter)- $EB(\epsilon)$	52
3.3.8	Onion-skin filter- $OS(r)$	54
3.4	Performance characteristics of choice filters	56
3.4.1	Filtering efficiency	56
3.4.2	Time complexity	62
3.5	Selecting a choice filter as a function of problem characteristics and user needs	64
4	DECISION SITUATIONS WITHOUT UNCERTAINTY	66
4.1	Introduction	66
4.2	Tolerances to model situational demands in the absence of uncertainty	66
4.3	The filter-choice problem for decision situations without uncertainty	68

4.4	Pragmatics of mapping choice filters to decision situations	69
4.4.1	ϵ -box Twopass filter	70
4.4.2	Superstrict dominance filter	72
4.4.3	DD-dominance filter	72
4.4.4	Onion-skin filter	73
4.5	A scheme for the filter-choice problem in the absence of uncertainty	74
5	DECISION SITUATIONS WITH UNCERTAINTY	
	PART I: BAYESIAN CHOICE FILTERS	77
5.1	Introduction	77
5.2	The MCDM-problem and uncertainty	78
5.3	Use of Bayesian Decision Rule(BDR) under Uncertainty	79
5.3.1	Producing the Bayesian Ordering	83
5.3.2	Summary	85
6	DECISION SITUATIONS WITH UNCERTAINTY	
	PART II: TOLERANCE BASED DOMINANCE FILTERS	87
6.1	Introduction	87
6.2	Performance of measured dominance for bounded noise models . .	88
6.3	Performance of measured dominance for unbounded noise models .	89
6.4	TBDFs for bounded noise models	92
6.5	TBDFs for unbounded noise models	97
6.5.1	Mapping conservatism expressions to tolerances	98
6.5.2	Bounding assurances based on expressions of conservatism .	102
6.5.3	Generalized nearly-Necessary and nearly-Sufficient sets: The In-Between Sets	106
6.6	Mapping user needs to near-sufficiency and near-necessity	109
6.7	Relating In-between sets to Pruning demands	111
6.7.1	A procedure to produce an ordering of In-between sets . . .	115
6.7.2	Reducing the computational complexity of producing the or- dering of in-between sets	118
6.8	Performance of Choice Filters under Uncertainty	121
6.8.1	Deriving an expression for the expected number of false pos- itives and false negatives of a choice filter	122
6.8.2	A Complexity analysis of computing the expected number of false positives for a choice filter	124
6.9	Monte Carlo Simulations to assess performance of choice filters . .	125

6.9.1	Data Generation	126
6.9.2	Simulations	126
6.9.3	Obtaining performance for various parameterized instantia- tions of a filter	127
6.9.4	Estimation of Confidence Intervals	127
6.9.5	Results	128
6.10	Mapping uncertain situations to choice filters	133
6.11	Summary	138
7	CONTRIBUTIONS AND FUTURE WORK	140
7.1	Contributions of the Dissertation	140
7.2	Future Work	143
7.2.1	Decision situations with very large number of criteria	144
7.2.2	Representing uncertainty in the Viewer	145
7.2.3	The Filter as an adviser/critic	145
APPENDIX		
A	PROOFS OF THEOREMS AND COROLLARIES	147
A.1	Introduction	147
A.2	Proofs for Chapter 2	147
A.3	Proofs for Chapter 3	148
A.4	Proofs for Chapter 6	158
BIBLIOGRAPHY		165

LIST OF FIGURES

FIGURE		PAGE
1	The Seeker-Filter-Viewer architecture	15
2	Fraction survivors produced by the dominance filter with varying n and m	26
3	A plot of expected fraction of survivors for the dominance filter with varying values of the correlation coefficient.	32
4	A plot of expected number of survivors for the dominance filter with varying values of the correlation coefficient.	33
5	A geometric representation for a 2 criteria problem to compute the survival probability of a random alternative X for various TBDFs. . .	57
6	Plots showing fraction survivors with varying number of criteria for four different choice filters when applied to a decision problem with 2500 alternatives and with tolerances set to 0.01 and 0.1 respectively. . .	61
7	The filter-choice problem for situations without uncertainty.	69
8	The filter-choice scheme for situations without uncertainty.	75
9	The Gaussian Noise Model	80
10	An illustration of the sufficient, necessary, and an in-between set of survivors corresponding to different classes of decision-makers.	95
11	Comparing two distributions in the location family.	98

12	Relating nearly-necessary (ϵ) and nearly-sufficient(ϵ) sets to measured dominance	108
13	An ordering to produce in-between sets	117
14	Overall complexity for ordering the in-between sets with new algorithm	120
15	FP-FN characteristics for the Strongly-Strict–MPS–Superstrict family from Monte Carlo simulations for various values of n , m and σ	129
16	FP-FN characteristics for the in-between sets obtained from various pruning expressions of the DM from Monte Carlo simulations.	131
17	TP-TN characteristics for the in-between sets obtained from various pruning expressions of the DM from Monte Carlo simulations.	132
18	A plot showing Student T-interval 95% simultaneous confidence estimates for each of the points in the FP-FN curve	134
19	The filter-choice scheme for uncertainty situations with bounded noise models	136
20	The filter-choice scheme for uncertainty situations with unbounded noise models	137

CHAPTER 1

INTRODUCTION

1.1 Decision-making and Decision Support Systems

Decision-making involves choice from within a set of alternatives, which may be finite or infinite in number, available either explicitly or implicitly as choice alternatives. Certain outcomes are desirable to the decision-maker. The decision-making process should therefore select alternatives that have the greatest chance of maximizing the outcomes desirable to the decision-maker. Various sorts of decision support can be provided to aid the decision-maker - the identification and generation of the choice alternatives in case the alternatives are not available explicitly, the identification of the attributes of the choice alternatives that one should care about, and in getting the values for these attributes for the choice alternatives at hand. Hence, some aspects of decision support can be computational in nature while other aspects might involve providing knowledge-based support.

If the alternatives are explicitly available or have been generated and once the attributes of the alternatives which are of importance to the decision-maker have been identified, the task of decision support is one of helping the decision-maker select the *best* alternative(s) from the available set. Even in the presence of a finite set of alternatives, human decision-makers, due to their limited computational capacities, have been known to find the task of choosing the best alternative to be daunting. As stated by Tversky [54], “in choosing among many complex alternatives such as new cars or job offers, one typically faces an overwhelming amount of relevant information. Optimal policies for choosing among such alternatives require involved computations based on weights assigned to the various relevant factors, or on the compensation

rates associated with the critical variables. Since man's intuitive capacities are quite limited, the above method is quite difficult to apply." In such circumstances, humans have been known to use various decision heuristics to solve the decision-making problem so as to strike a reasonable balance between the complexity of the problem and the resources at hand ([50], [52], [52] and [40]). As shown in [54], the use of a decision heuristic will not always lead to the best choices from the viewpoint of the decision-maker. Hence a very important dimension of decision support is the provision of computational support to the decision maker so that the employment of suboptimal decision heuristics is curtailed or at least limited.

1.2 Multiple Criteria Decision Making (MCDM) as the norm

The norm has been to view most decision problems as belonging to the class of multiple-criteria decision-making or MCDM, the word *criteria* being used to refer to the attributes of the choice alternatives that have been identified as of interest to the decision-maker. More formally, the MCDM-problem is:

$$\text{maximize } \{d_{i1}, d_{i2}, \dots, d_{im}\} \text{ for } d_i \in D$$

where D is the set of n choice alternatives and d_{pq} is the value taken by the alternative d_p on criterion q .

A standard technique directed towards producing the *best* solution for the MCDM problem is to produce one, some, or all elements of the Pareto Optimal set of alternatives from the given set; the best solution then is the alternative(s) within this set which best satisfies the decision-maker. One way to produce the Pareto Optimal set is to apply the *principle of dominance*. This principle makes use of the *dominates* relation to declare the dominated alternatives to be suboptimal; the *dominates* relation is defined as below:

Dominance: Multicriterially evaluated alternative A is said to dominate another multicriterially evaluated alternative B if A is at least as good as B in all the criteria and there is at least one criterion in which A is strictly better than B . i.e.

$$(A \text{ dominates } B) \iff \exists i : (A_i > B_i) \wedge \forall j : (A_j \geq B_j),$$

where X_k refers to the value taken by alternative X on criterion k .

The Pareto Optimal set is also referred to, in literature, as the *Efficient set*. The idea of a Pareto Set applies to a discrete alternative MCDM problem as defined above for which the number of decision alternatives is finite. For decision problems where the number of alternatives is potentially infinite in number, one refers to the Pareto Surface rather than the Pareto Set. Under such circumstances, and as mentioned earlier, an extra dimension of the decision support might involve either generating some of the Pareto Optimal points, or generating a finite sample from the potentially infinite number of alternatives and obtaining the Pareto Optimal set from it. For purposes of the dissertation, we consider only discrete alternative MCDM problems.

In most real world decision problems, it will be rare that a single alternative will excel in all the criteria. This is because, typically, criteria often conflict with one another and alternatives which evaluate to a good score on one criterion would not do so well on some other criteria. This results in a Pareto Optimal set that has more than a single alternative. In such a case, the Pareto Optimal set is known to have the property that, for any alternative in the set, the only way to improve along any criterion is to accept a loss in some other criterion. Therefore, choice among Pareto Optimal alternatives becomes a matter of making trade-offs.

There are many possible ways to handle the problem of choosing from Pareto Optimal alternatives. For decision problems where any alternative will do so long as it is Pareto Optimal, one of the alternatives from the Pareto Optimal set can be randomly chosen as the final solution. For problems where this is not the case, the problem again comes down to choosing from within a set of alternatives.

Different techniques address this problem by use of different mechanisms. For example, one class of techniques make use of weights to extract the final solution. The decision-maker expresses weights or weighting coefficients for the individual criteria such the weight values are in proportion to the relative importance of each criterion to him. These weights can now be used to select Pareto Optimal alternatives of interest to the user.

Another class of techniques elicits the user's trade-off expressions for each pair of criteria in the form of *marginal rates of substitution or MRS*¹ and uses these expressions to obtain the final solution. Studies, however, indicate that the use of concepts like weights and MRS is suspect due to various reasons, ranging from whether or not decision-makers can relate to these concepts, to whether an accurate translation of the user's desires to some of these concepts is even possible (see [7],[20],[37], [19], [22], [23]). Hence, it is not clear if the solution produced thereby is indeed the best solution with respect to the decision-maker, given that these concepts might not represent his desires completely and accurately.

In the next section, we discuss an approach that views the decision-making problem as one of modeling the decision-maker as a holder of different kinds of preferences. The presence of different kinds of user-preferences motivates the requirement that DSSs must be sensitive to the elicitation of each of these kinds of preferences by means which are natural and easy for the decision-maker to comprehend. By virtue of this sensitivity, the DSS will be shown to not only facilitate accurate expression of preferences by the user, but also to be responsive to different kinds of decision situations

1.3 Modeling the Decision Maker

One approach to providing support for a decision problem involves modeling the decision-maker as a holder of various kinds of preferences. Psychological literature in decision-making uses the term *preferences* to characterize the values of a decision-maker that are used in selection from among a set of alternatives. Decision support is seen as eliciting these preferences from the decision-maker in a natural way and applying these preferences to the given set of alternatives. For computational assistance in the decision-making, it would be attractive if a decision-maker's preferences can be expressed in such a way that a DSS can use them to select alternatives on behalf of the decision-maker. If the preferences have been accurately and completely elicited, the decision-maker can be assured that the decision alternatives that he gets

¹The MRS for a pair of criteria is the amount of decrement in the value of one criterion that compensates the decision-maker for the one-unit increment in the value of the other criterion, while the values of all other criteria remain unaltered.

with the assistance of the DSS are the same alternatives that he would have picked in the absence of the DSS.

Some preferences are *context-independent* i.e, they are independent of the specific alternatives. For example, prior to obtaining the Pareto Optimal set, the criteria to be used for dominance along with their directions of goodness, can be elicited from the decision-maker without reference to the set of alternatives at hand. We call such preferences *abstract preferences*. There are the following advantages to the use of abstract preferences:

- They make it easy to automate the decision-making process. The abstract expressions can be applied by an automated system like a computer to the set of alternatives at hand to produce the final solution.
- Since these preferences are independent of the alternatives at hand, the decision-maker does not have to be burdened with the size of the problem. In other words, the number of alternatives at hand will not directly affect the amount of effort, time, and information-processing, that the DM has to incur in order to express these preferences.

Unfortunately, studies in behavioral decision sciences indicate that not all preferences can be expressed abstractly by the user. Tversky [53] shows that choice among options may be context dependent; the relative value of an option depends upon not only on its own characteristics, but also upon characteristics of other options in the choice set. As stated by Tversky, “people do not maximize a precomputed preference order, but construct their choices in light of the available options.” The expression of such context -dependent preferences is often easily done in the presence of concrete alternatives at hand. We refer to such preferences that are expressed in terms of concrete alternatives at hand as *concrete preferences*. Tradeoff expressions are often better expressed as concrete preferences, although sometimes tradeoffs can be expressed independent of the alternatives as in, “a 5% increase in cost for a 10% increase in functionality is acceptable.” However, it is often the case that tradeoffs are highly nonlinear in the space of alternatives, and in any case the decision-maker may not be able to articulate his tradeoffs except when presented with the specific

alternatives. Therefore, for many problems it might be inevitable that preferences be elicited as concrete preferences.

The expression of concrete preferences has its own set of problems as shown by Tversky [54]. In the presence of an information overload, say due to a large number of alternatives, people use heuristics to accommodate the mismatch between their limited cognitive capacities and the information processing warranted. As a result, the choices produced by the application of such preferences cannot be guaranteed to be the best choices from the viewpoint of the decision-maker. Moreover, since the expression of concrete preferences is completely guided by the decision-maker's interaction with the actual data at hand, the solutions produced might be very sensitive to the path taken by the decision-maker which is a function of what catches the eye of the decision-maker first.

A third kind of preferences come about by virtue of the constraints that the decision-maker wishes to impose upon the decision-making process itself. For example, if the decision-maker is under a lot of time-pressure, he might want a quick solution. Note that *quickness* in decision-making is a criterion that has nothing to do with attributes of the decision alternatives. However, inability to cater to such preferences can render the DSS ineffective. For example, the decision-maker described above might be open to satisficing (nearly optimal) solutions as long as the decision-making is quick. A DSS which produces an optimal, but not a quick, solution might not be of much utility to the above user.

Therefore, if providing support for decision-making is considered as a matter of eliciting and applying the decision-maker's preferences, there are at least two classes of preferences that need to be catered to - preferences that have to do with the decision problem (we call them *object preferences*), and preferences that have to do with the decision-making process (what we call *process preferences*). Within the former class, some preferences (like criteria that matter to the DM) are better elicited abstractly and are called *abstract preferences* while other preferences (like trade-offs) are mostly easier to express after considerations based on the choice alternatives at hand; these are *concrete preferences*. In general, there can be a conflict between some of the *object preferences* and the *process preferences*. In such a case, the decision-maker will need to be told of his unrealistic expectations and presented with ways in which

he can relax some of these preferences in order to obtain a set of preferences that is mutually satisfiable. We will soon present a decision support architecture which is based upon the above view of providing decision support. But first, we present a brief survey of some of the existing MCDM techniques.

1.4 Existing approaches for solving the discrete alternative MCDM problem - A Brief Survey

The MCDM literature identifies many techniques for solving the discrete alternative MCDM problems. Zionts [47], Despontin [30], MacCrimmon [26], Hwang and Yoon [14] are some references providing surveys describing and evaluating these techniques. Ozernoy [34] estimates that there are at least 50 available MCDM methods [pp. 163].

As stated previously, in a discrete alternative problem the Pareto set is obtainable by the use of the dominance principle. However, since the Pareto set can itself contain multiple alternatives, the decision problem remains unsolved until choice is made from this set (unless of course the entire Pareto set is of interest to the decision-maker). Therefore, most techniques are focused on obtaining preference information from the decision-maker that will not just enable the obtaining of the Pareto set but also make it possible to produce only a handful of alternatives as the final solution. A few techniques do this by trying to produce only a subset of the Pareto set, while a few others are not committed to producing only Pareto Optimal alternatives as final solutions. In recent years, however, it has become general consensus that the final solutions must be from the Pareto set- each of [35], [36], and [29] state that the *final* solution of the MCDM problem should be the one which is the most preferable to the decision-maker relative to any of the Pareto optimal alternatives, and therefore that the final solution(s) must be Pareto optimal.

Hwang and Yoon [14] classify discrete alternative MCDM methods according to the *form* and the *depth* in which a decision-maker's preferences can be expressed. For instance, the form of preference expression can be *attribute-based* (preferences are expressed in terms of the criteria) or *alternative-based* (preferences are expressed in terms of pairwise choices made by the decision-maker). The latter class of preferences shares some properties with what we identified as *concrete preferences* previously, but

it really allows for context-dependent preference expression only in the presence of pairs of alternatives. In general, the context dependency that we referred to with respect to concrete preferences includes the presence of all other competing alternatives and not just pairs. Within each of these two forms of preference expression, Hwang and Yoon [14] further classify methods based on the depth, or degree, of preference expression. For example, attribute-based information could be either in the form of criteria-thresholds, ordering of criteria, weights for criteria, and so on. We next describe the various techniques as discussed in [14].

If only the criteria of interest to the user have been identified and no other preferences are expressed by the user, then popular techniques include the maximin and the maximax techniques. The maximin and maximax techniques apply only if the criteria are commensurable or have been normalized. The maximin technique involves finding for each alternative, the criteria on which it performs the worst, and in the end selecting the alternative which has the most acceptable evaluation from these evaluations. As stated in [14], this technique is ultra-pessimistic in that, for each possible choice, it considers only the worst that can happen, while ignoring other criteria evaluations of the choice. Also, this technique can produce a final solution that is not in the Pareto set. On the other hand, the maximax method obtains for each alternative its highest criteria evaluation and then selects the alternative that has the maximum such evaluation. Provided there are no ties on the highest criteria evaluations, this technique can be expected to produce a Pareto optimal solution. However, alternatives that perform reasonably well across all criteria will be ignored in favor of alternatives which are virtuous along a single criterion alone by the maximin and the maximax methods. One would expect these multicriterially optimal alternatives to be of interest to most decision-makers, if not preferable to the unicriterially optimal ones. This technique can be considered as ultra-optimistic, in the sense that it satisfies a decision-maker who is interested in the alternative for which the best criteria evaluation is the best from the remaining, while ignoring other aspects of the alternative.

The next class of methods take additional information related to the criteria of interest to the user. This class includes several subclasses of techniques.

- Techniques like the conjunctive method and the disjunctive method take criteria

information in the form of criteria-thresholds expressed by the decision-maker. In this subclass, the user is told to specify minimal acceptable values along each criterion. In the conjunctive method, only alternatives which are better than the minimum acceptable values along all the criteria are retained as final solutions. Therefore, alternatives which perform well on all but one criterion will be eliminated as well. In general, very few alternatives can be expected to excel along all criteria and therefore the application of the conjunctive method with very high thresholds might result in none of the alternatives surviving as solutions. One way to counter this effect is to use the disjunctive method in which an alternative has to only score better in any one criterion relative to the minimum acceptable value for that criterion in order to be retained as a final solution. However, this can result in a large number of alternatives surviving as final solutions. Clearly, both techniques will produce final solution sets that need not contain only Pareto Optimal alternatives. Also, they produce multiple solutions and often choice from this set involves picking up a random alternative from the set and presenting it to the user. In such a case, the solution cannot be guaranteed to be optimal.

- Techniques like Lexicographic Ordering and Elimination By Aspects take criteria information in the form of the ordinal preference or ranking of the criteria by the user. In lexicographic ordering, the user is expected to order all the criteria according to their relative degree of importance to him. Alternatives are now selected starting with the most important criteria and retaining the best alternative along that criterion; if there are multiple survivors, the process is repeated for the survivors using the second-most important attribute and so on until a single alternative survives. This technique does not allow for a small increment along an important criterion to be traded off with a great decrement of a less important attribute while such a trade off might often be appealing to the decision maker. To counter this, the technique is modified as lexicographic semi-ordering wherein apart from ordering the criteria, the user is also expected to specify bands of imperfect discrimination along the criteria. When alternatives are considered along some criterion, not only those alternatives which perform the best in the criterion are retained, but additionally

alternatives which are not significantly or noticeably different along that criterion relative to the highest alternatives as also retained before processing along the next-most important attribute. Therefore, alternatives which do not score the highest along the most important criterion but which do reasonably well on it will be retained for comparison during the next stage. Tversky [54] shows that this modified technique can be shown to lead to intransitive choices. Both techniques can however be shown to produce only Pareto optimal solutions. Elimination by aspects is another technique introduced by Tversky [55] which belongs to this subclass. In this method, each alternative is viewed as a set of aspects where the aspects might be either criteria or they could be arbitrary features of the alternatives. Tversky's model now describes choice as an elimination process governed by successive selection of aspects instead of cutoffs. It is similar to the lexicographic techniques but it differs from them in that, the attributes are not ordered in terms of importance, but in terms of their discrimination power in a probabilistic mode. Due to its probabilistic nature, the criteria for elimination and the order in which they are applied vary from one occasion to another and are not determined in advance. This technique can be shown to produce only Pareto optimal alternatives. However, as Tversky himself noted, this technique can lead to the elimination of alternatives which can be better (multicriterially) than those which are retained.

- Techniques which take the user's criteria-based preferences as cardinal information either in the form of weights or utility functions, concordance measures, or ideal points. The methods belonging to this class all involve implicit trade-off but they differ in terms of how this achieved:
 1. The final solution is an alternative with the largest utility expressed using some kind of weighting technique, with weights specified by the user to express intercriteria preference. In simple additive weighting, the decision-maker is expected to supply importance weights to each criterion so that the weights sum to 1. Each alternative is now rated as a linear combination of its criteria values with the corresponding weights. The alternative with the highest rating is chosen as the final solution. As mentioned in

[7], it is often difficult to understand what the weights actually represent. Hobbs [20] states that instead of relative importance they must be held to represent the trade-off expressions of the decision-maker. Nakayama [22] mentions indicates that it is difficult to control the direction of the solutions by the weighting coefficients. Mietinnen [33] states that small changes to the weight vector may cause big changes in the quality of the solutions; at the same time, dramatically different weight vectors can produce quite similar solutions. In Das and Dennis [15] it is emphasized that an evenly distributed set of weighting vectors does not necessarily produce an even representation of the Pareto Optimal set. Also, in continuous MCDM problems, the weighting technique cannot produce the Pareto Optimal points that lie in the non-convex regions of the Pareto surface.

2. The alternatives are ranked to achieve an overall preference ranking which best satisfies a given *concordance measure* as in ELECTRE [4], [2], [5],[3],[32]. The ELECTRE family of methods uses an outranking relationship that is built on two indices, the concordance index and the discordance index. Based on these indices, it can be ascertained for any pair of alternatives if either outranks the other using their criteria values. This technique also requires the use of criteria weights as in (1) above. Also, the user is expected to furnish the two indices by translating it in terms of his risk propensity. Also, [14] states that the two indices are rather arbitrary although their impact on the final solution can be significant.
3. The alternative which has the largest relative closeness to the ideal point is chosen as the final solution. Here, the ideal point refers to the maximum value on each criteria considered as a single criterion vector. For most decision problems, this point does not pertain to a single alternative. The techniques in this subclass try to minimize the Euclidean distance from the ideal point and simultaneously maximize the Euclidean distance from the negative ideal point (criteria vector composed of the minimum criteria values) to obtain the final solution. For some problems, the simultaneous achievement of these two goals becomes impossible. The user is expected to provide information in the form of weights expressing his

relative importance of the criteria.

- Other techniques take tradeoff expressions of the user in the form of marginal rates of substitution. This was already mentioned previously as a technique whereby the user expresses his trade-off expressions along all pairs of criteria in terms of the marginal rates of substitution (definition in Section 1.2). This expression can be used to choose fewer points from the Pareto set. The problems associated with this technique was also discussed in Section 1.2.
- Techniques which take preferential information in the form of a set of choices made by the user when presented with pairs of alternatives. It is expected that the set so obtained may contain inconsistent choices. The methods belonging to this class produce their final ordering by trying to minimize the inconsistency between the ordering and the set of choices expressed by the decision-maker. The output is typically a set of weights that is supposed to best reflect the choice made by the decision-maker. If the number of alternatives is large, the large number of overall pairwise choices, which is $n(n - 1)/2$, can be cumbersome for the user to interact with and there is a need to select a salient set of pairs that best reflect the entire space of alternatives in the original set.

This finishes the brief survey of discrete alternative MCDM problems. As stated by Ozernoy [34] the different methods represent radically different approaches to decision-making. The choice of an appropriate method for a problem itself becomes an MCDM problem. The properties along which these individual MCDM methods can be evaluated are many, including the kinds of information that are available about the problem as well the decision-maker's preferences. In the next section, we introduce the term *decision situation* to capture all such properties and more which can be used to decide on the best technique to solve a given problem in a given decision situation.

1.5 Decision Situations: The need for situation-specific decision support

Many of the currently existing techniques for solving MCDM problems leave out issues that are important to practical DSSs. This is mainly because most such techniques are focused towards solving the decision problem independent of the situation under which the problem is being solved. For example, consider a situation wherein the decision-maker desires a quick, satisficing, single alternative solution. In the above specification, *quickness*, *satisfiability* and *unity* are all properties which are independent of the decision alternatives. In fact these are properties related to the decision-making process, and what we referred to earlier as process preferences. The above process preferences are representative of a class of users who are in high time-pressure to make a decision, to whom optimality is not a primary concern and for whom pruning to a single final solution is necessary. MCDM techniques that are unresponsive to such demands of the user will render the resultant decision support ineffective. Therefore, it is important that DSSs be designed so that they can provide situation-specific support.

The process preferences expressed by the decision-maker characterize only part of the decision situation. The decision situation also includes the characteristics of the problem at hand. For example, if the criteria values obtained for the alternatives are noisy or cannot be trusted to be accurate, the DSS must still be able to produce a solution that is robust even in the presence of noisy criteria evaluations, or for which assurances of different sorts can be given. For example, in the presence of noise, it may be possible to give guarantees about closeness to optimality, if not optimality itself; this can make the solution completely acceptable to the certain decision-makers. In the dissertation, we refer to *decision situations* as composed of the process preferences of the decision-maker and the characteristics of the problem at hand.

1.6 A Decision Support Architecture for MCDM

In the preceding sections of the chapter, we analyzed some existing techniques for solving the MCDM problem. Also, using various results from behavioral sciences, we

were able to state some desirable properties for decision support systems. To restate these requirements, we need DSSs to possess the following desirable properties:

1. Decision Support Systems must support the elicitation of object preferences (concrete and abstract preferences) as well as process preferences of the decision-maker by means which are natural. Additionally, DSSs must be responsive to the various characteristics of the decision problem at hand.
2. Decision Support Systems must augment the computational capacity of the decision-maker such that the solution(s) produced by the decision process respects all of the preferences expressed in (1) to the degree feasible.

In the next section, we will look at the S-F-V-architecture[42], which was developed at the OSU-LAIR² and examine its applicability as a decision support architecture in the context of the above requirements. One of the primary goals of the dissertation will be to investigate ways in which the S-F-V architecture can be modified so that it is sensitive to various decision situations and thus applicable to a large class of decision problems.

1.7 The Seeker-Filter-Viewer Architecture

In [42], Josephson, Chandrasekaran et. al. propose an architecture, the *Seeker-Filter-Viewer (S-F-V)* architecture, for the exploration of large design spaces. In this paper, design optimization is viewed as being essentially multicriterial. In this section, the three different modules of the architecture will be considered one at a time. This will provide us with a better context to analyze how the architecture meets the requirements stated as desirable above. The primary components of the S-F-V-architecture, along with the human in the loop, are shown in Figure 1 below.

²The Ohio State University - Laboratory for Artificial Intelligence Research

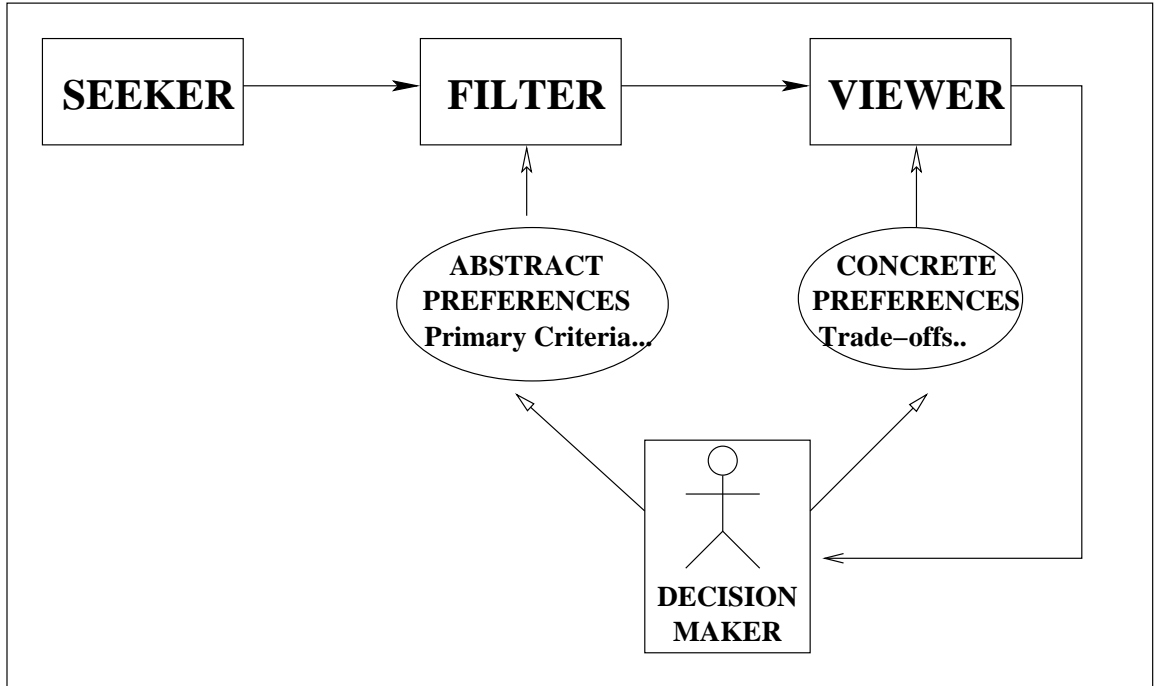


Figure 1: The Seeker-Filter-Viewer architecture

1.7.1 The Seeker

The Seeker is responsible for generating, or otherwise making available, the different alternatives and their evaluations along the various criteria, i.e., it generates the $C(D)$ matrix for the MCDM formulation stated in Section 1.2.

1.7.2 The Filter

The Filter allows the DM to express his abstract preferences. From among the available attributes of the alternatives, the decision-maker is allowed to choose those which directly relate to his main concerns towards obtaining desirable outcomes. For example, in a printer purchasing problem, if the decision maker is looking for a printer to use on a daily basis, to print a large number of documents, and for which the quality of the printouts need not be very good, then attributes like long-life, reliability and high speed of printing might be of primary interest to the user while other attributes like resolution, and color-printing need not interest him.

The indicated attributes, of interest to the user, are used as the dominance criteria by the Filter to produce the Pareto Optimal set; hence such attributes are referred to as the *primary criteria* with respect to the user. In other words, the primary criteria are the criteria in terms of which the decision-maker associates the notion of optimality of an alternative. In many cases, the problem of choosing the primary criteria might itself require the provision of decision support to the decision-maker. This is because many times, the relation between the underlying criteria of importance to the user and the attributes of the alternatives which map to those criteria might not be directly visible to the decision-maker. In such a case, the DSS must be able to use the primary user-concerns and choose the primary criteria on its own. Currently, the S-F-V architecture does not provide any support in the selection of the primary criteria. A list of all evaluated attributes pertaining to the choice alternatives is presented to the user and he is expected to choose the primary criteria on his own. By virtue of the fact that most attributes that are used often used as primary criteria are only tokens of the underlying criteria of importance to the user, these attributes are often called as *proxies or proxy criteria*. In the use of proxies to represent criteria, one needs to be careful in ensuring that the correspondence between the criteria being modeled and the proxy holds uniformly across the entire range of the proxy values. The alternatives that remain after the application of the Filter will be referred to as the *survivors* of the Filter.

As the number of primary criteria increases, the Filter's pruning efficiency decreases. Thus when the number of criteria is large, it is essential that only the attributes of primary interest should be selected by the decision-maker as the primary criteria. Those attributes which are of interest to the decision-maker but which he doesn't consider to be primary are referred to as secondary criteria. Secondary criteria are typically meant to be used by decision-maker to help in breaking a tie between two Pareto Optimal alternatives. In the S-F-V architecture, this is done at the Viewer stage, when the user interacts with the survivors of the Filter to express his tradeoffs and other concrete preferences.

In summary, the Filter elicits and applies the abstract preferences of the decision-maker to the initial set of alternatives. The primary advantage of filtering based on the dominance principle is that provided all the primary criteria have been correctly

identified by the decision-maker, none of the eliminated alternatives can be of interest to the decision-maker since there is an objective justification or rationale for the elimination of such alternatives - that each such alternative is *dominated* by some efficient alternative. Thus, the Filter helps in reducing the size of the problem, thereby reducing the amount of information that needs to be dealt with by the decision-maker at a later stage. Provided all the primary criteria have been correctly identified before Filtering, there is no loss incurred by the decision-maker due to the elimination of alternatives by the Filter.

1.7.3 The Viewer

If all of the decision-maker's preferences regarding the alternatives are completely expressed at the Filter stage, then a random element of the Pareto Optimal set should satisfy the decision-maker and nothing further needs to be done. However, in most problems, a decision-maker is likely to have preferences in addition to those identified as criteria of interest at the Filter stage. The Viewer provides means by which the decision-maker can express additional preferences to choose from among Pareto Optimal alternatives. As mentioned earlier, choice from among Pareto Optimal alternatives involves making tradeoffs, and in many problems the tradeoff preferences tend to be nonlinear in the space of alternatives. Moreover, in many cases it becomes easier for the decision-maker to express his tradeoffs as concrete preferences (namely in the context of the existing options). The design of the Viewer is based on the above observations.

Interaction with the Viewer involves use of various kinds of interactive diagrams like 2-D plots, range plots and bar-charts to express concrete preferences among the Pareto Optimal alternatives. By means of the graphical user interface provided by the Viewer, the decision-maker chooses regions or alternatives of interest to him by making selections on any of these diagrams. An example of such an interactive session between the user and the Viewer can be found in [13][pp 5-8]. These selections might be made by the user based any of the following observations:

1. The selected alternatives lie on a region in a 2-D plot so that they are the best along both the criteria which form the axes of the 2-D plot. Of course, since

all the alternatives being viewed are Pareto Optimal, it is to be expected that these alternatives would have to necessarily lie on inferior regions along some of the other criteria. All selections made by the decision-maker on a single plot are cross-linked across the remaining plots, so the decision-maker can see the extent to which the selected alternatives perform poorly in the other criteria. If the inferiority in the other criteria happens to be within a tolerable level from the viewpoint of the decision-maker, then he can use the Viewer to narrow down to the set of alternatives at hand, with the remaining alternatives being removed from present consideration. Now the decision-maker might use some other plot to choose from among the survivors of the first selection and so on.

2. The selected alternatives lie in a region in a 2-D plot such that their superiority along one of the criterion-axis greatly outweighs or compensates for (according to the user's tradeoff preferences) the extent to which they are inferior along the second criterion axis. Again, the user proceeds in the manner described in (1) above.
3. The selected alternatives might lie in an isolated region on some plot relative to the other alternatives. In such a case, a decision-maker like a designer might select these alternatives merely to examine the reasons that these few alternatives have isolated performance measures. To do this, the decision-maker might further pull up a plot showing the structural properties of the alternative and see if the isolated points form a separate cluster in some region of the structural space as well. This allows the decision-maker to test and even acquire knowledge about the domain by using the Viewer.
4. The selection might be based on a category that the decision-maker is interested in, like *two-door cars*. These selections are made in bar-charts which display the alternatives as belonging to separate categories.
5. Finally tradeoffs involving secondary criteria to break Pareto Optimal ties can also be done in the Viewer by pulling up an appropriate plot which shows the evaluations of the alternatives along the secondary criterion of interest to the user.

These are only some of the ways in which the decision-maker can use the Viewer to select from among the survivors of the Filter. Additional facilities supported by the Viewer include:

- retracting previous selections,
- retracing the path leading to a set of survivors arrived at by a series of Viewer-based selections,
- reverting back to the original set of alternatives,
- maintaining the distinction between two separate selections by use of different colors to highlight the alternatives in each of the two selections,
- combining sets of selections in various ways, and
- looking at the entire set of specifications for the alternatives in a selection, as a table.

By means of the above features, the Viewer allows for the natural expression of additional user-preferences in order to choose from Pareto Optimal alternatives.

1.8 Motivation, Goals, and Outline of the Dissertation

The Filter-Viewer combination in the S-F-V architecture imparts many of the properties that were stated in Section 1.6 as desirable for a DSS to be effective.

- The Filter elicits preferential information by means which are natural and easy for the decision-maker to comprehend. No such quantities like weights, marginal rates of substitution, are expected.
- The Filter augments the computational capacity of the DM by performing Dominance Filtering on his behalf. Studies, [49], show that people find the task of producing the Pareto set computationally demanding which results in the choice of suboptimal alternatives.

- The Viewer allows the decision-maker to express his tradeoffs. Since tradeoffs are typically easier and more common to express in the context of the existing alternatives, the Viewer is designed to aid the decision-maker in the expression of concrete preferences.
- The design of the Viewer is tailored around the easy and natural elicitation of concrete preferences. Thus the Viewer elicits those preferences which are not easy to express abstractly, or at the Filtering stage. Moreover, the Viewer allows for further pruning in case the Filter falls short of producing a handful and virtually augments the filtering efficiency of the Filter.

In summary, not only do the Filter and the Viewer provide the architecture with desirable properties but they also complement each other's shortcomings. This symbiotic relation between the two modules is highly instrumental in making the overall architecture effective for use as a decision support architecture.

However, the S-F-V architecture lacks the ability to provide situation-specific decision support. Consider a decision problem wherein the criteria values are known to be noisy. In other words, the actual values taken by the alternatives on the various criteria are not known; only approximate values are available along with bounds placed on the approximations. There is no mechanism either in the Dominance Filter or in the Viewer to account for the presence of noise in the criteria values. Ideally, we would want to modify the Filter so as to produce the set that would have the greatest chance of being the Pareto set based on the actual criteria values. However, since the actual values are not available, the application of dominance based on the measured values alone does not guarantee this. Clearly, the filter needs to take information about the noise associated with the various criteria into account.

The primary goal of the dissertation is to show how the S-F-V architecture can be tailored to become responsive to various decision situations. We do this by building a framework which instantiates the S-F-V architecture in various modes so that the instantiated architecture will retain all of the good properties of the original S-F-V architecture while at the same time will impart it with the flexibility to be responsive to various decision situations. More specifically, we formulate the notion of a Choice Filter, of which the dominance filter is one kind. A choice filter can be made to cater

to the abstract preference expressions of the decision-maker and also take into account the situation-specific parameters of the decision problem. The idea is to instantiate the S-F-V architecture with an appropriate choice filter towards making it responsive to the decision situation.

1.8.1 Goals of the Dissertation

The primary goals of the dissertation are:

- To explore the design space of Filters which are variations or extensions of the Dominance Filter so that a larger variety of user-needs and problem characteristics can be accommodated. The dimensions of variation for user needs include the degree of pruning desired, expressions related to the number of false positives and false negatives produced in the final set for decision problems with uncertainty, expressions of near-optimality to allow more alternatives than the Pareto set when more is desired. The problem characteristics investigated in the dissertation are uncertainty in criteria values, number of criteria and alternatives in the decision problem, presence of correlations among the criteria.
- To develop a body of analytical and experimental knowledge needed to create a framework for mapping decision situations to an appropriate instantiation of the S-F-V architecture towards providing situation-specific decision support.

1.8.2 Outline of the Dissertation

In Chapter 2, we analyze the performance of the Dominance Filter in terms of its complexity and filtering efficiency and their dependence upon several problem characteristics. Chapter 3 introduces the idea of tolerance-based dominance filters as choice filters; various choice filters will be defined and their performance characteristics will be analyzed. Chapter 4 considers decision problems without uncertainty and presents a scheme to map decision situations to various kinds of choice filters in the absence of uncertainty. Decision situations with uncertainty are dealt with in Chapters 5 and 6- Chapter 5 deals with choice filters based on Bayes Decision Theory while Chapter 6 examines the use of tolerance-based dominance filters for dealing with uncertain

decision situations. Chapter 6 also summarizes the results of Chapters 5 and 6 and presents a framework based on these results for mapping decision situations to choice filters. Chapter 7 presents a summary of the primary contributions of the dissertation, and some open questions which are orthogonal to its goals but seem like interesting questions to take up, as future research.

CHAPTER 2

PERFORMANCE ANALYSIS OF THE DOMINANCE FILTER

2.1 Introduction

In this chapter, we examine how the performance of the Dominance Filter varies with the number of alternatives n , the number of criteria m , and under different correlations among the criteria. The performance dimensions of interest are the Filter's *filtering efficiency* and *computational time complexity*.

2.2 Performance of the Dominance Filter

The performance analysis of the Dominance Filter presented in this section is based on existing results from [12], [27], and [9]. Before we present the analysis, we first define each of the two dimensions of performance of the Filter.

Definition 2.1 *The filtering efficiency η of a choice filter is the fraction of the total number of alternatives eliminated by the choice filter.*

Therefore, the fraction of survivors produced by the filter, which is the fraction of the total number of alternatives which survive the filter, is $1 - \eta$.

Definition 2.2 *The computational time complexity τ of a choice filter is the number of steps taken by an implementation of the choice filter to produce its survivor set.*

Therefore, the lower the time complexity of an algorithm, the better its performance in this measure. We will use the *order* notation to indicate the average and worst-case time complexities.

2.2.1 Filtering Efficiency of the Dominance Filter With Uncorrelated Criteria

The expected number of survivors of the dominance filter are obtained in both [9] and [12]. In [12] (pg. 143), the recurrence relation for the expected number of survivors upon dominance filtering with n alternatives and m criteria is given by the following approximation,

$$a(n, m) \approx \sum_{k=0}^{m-1} \frac{\ln^{m-k} n}{(m-k)!} + (1 - \gamma) \ln^{m-2} n / (m-2)! \quad (2.1)$$

where $\gamma = 0.5772$ is Euler's constant.

From this it can be seen that the expected number of survivors is polynomial in $\ln(n)$, with the degree of the polynomial depending upon the number of criteria, m . This provides the basis to arrive at [9]'s relation of,

$$a(n, m) = O(\ln^{m-1} n). \quad (2.2)$$

This indicates how the fraction of survivors from the original number of alternatives changes with respect to changing values of n and m :

- Since m appears in the exponent, the expected number of survivors will increase much faster as m increases (for a fixed n) than with an increasing n . This is because as m increases, it becomes more and more difficult for an alternative to be dominated. It is proved in Theorem 2.1 that all survivors of an m -criteria dominance will also survive an $(m+k)$ -criteria dominance (with k newly added criterion to the previous set of m criteria), assuming that no two alternatives are coincident along all criteria.

Theorem 2.1 *Every distinct³ survivor of an m -criteria dominance is also a survivor of $(m+k)$ -criteria dominance, $k \geq 0$, where the $(m+k)$ -criteria dominance uses all the criteria used by the m -criteria dominance.*

Proof: See Appendix A.

- In general, the larger the number of alternatives n , the larger the value of m required for all alternatives to end up as survivors.
- As n increases, the absolute number of survivors tends to increase.
- The filtering efficiency tends to increase as n increases. In other words, the fraction of survivors reduces as n increases. This is indicated in Figure 2. From the figure we can see that for any given value of m , the larger the value of n , the smaller the fraction survivors produced by dominance. This indicates that although the absolute number of survivors increases with an increasing n , the filtering efficiency of the dominance filter actually tends to improve. This property is justified from the expression for the expected number of survivors and the fact that n grows faster than a polynomial in $\ln n$. Also, geometrically the survivors of dominance in m -criteria space form the $(m - 1)$ dimensional Pareto surface. Intuitively, as n increases, one would expect a greater proportion of the newly introduced alternatives to fall *in* the m -dimensional volume contained by the Pareto surface, than *on* the $(m - 1)$ dimensional surface itself. This also explains why the fraction of survivors reduces as n increases.

Implications of the Filter's Efficiency for Decision Making

Analysis of the filtering efficiency of the Dominance Filter indicates that the large- m , small- n decision situation is the worst scenario for the application of the Filter since one would expect almost every alternative to survive. The application of the Filter, independent of how few alternatives it eliminates, is still defensible because the elimination of suboptimal alternatives prevents the decision-maker from being exposed

³By a distinct alternative is meant one for which no other alternative is coincident with it on all criteria

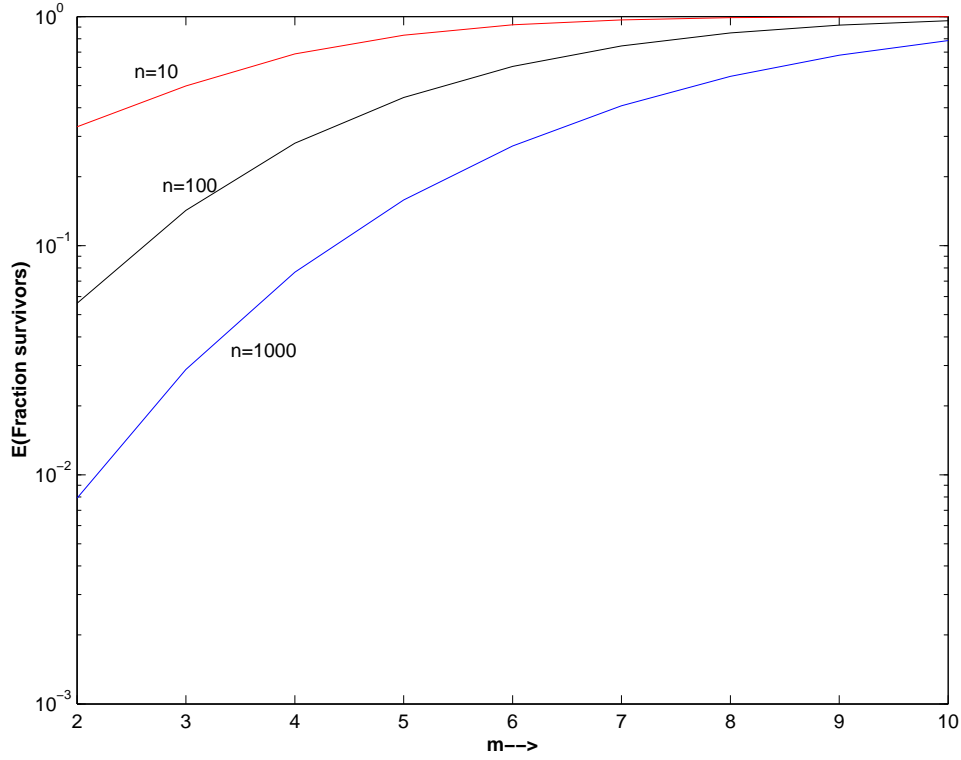


Figure 2: Fraction survivors produced by the dominance filter with varying n and m .

to choosing a suboptimal alternative at a later stage of the decision-making. Also, large- m situations are worrisome in general because the fraction survivors increases quite rapidly with increasing m and ending up with a large number of survivors might not be desirable for some decision situations. Of course, the Viewer can be used to further prune the survivor set. However, the number of plots in the Viewer which the user will need to refer to, before making selections and decisions based on selections, increases rapidly with increasing m as well. Thus, the cognitive load on the decision-maker increases during the Viewing-stage as well with an increasing m .

2.2.2 Computational Time Complexity assuming uncorrelated criteria

We use τ_{max} to represent the worst-case time complexity, and τ to represent the average time complexity of an implementation of the Filter. At first glance an implementation for computing the dominance survivors might seem to require all possible pairwise comparisons, which would make the algorithm $O(n^2)$ in the best-case, clearly not very good. However, as noted in [12], one can exploit the transitivity of the *dominates* relation to eliminate an alternative the moment it is dominated by another alternative. A pseudocode describing an implementation of the dominance filter is given below:

```

Dominance( $X(n, m)$ ){
   $U = X; S = X(1);$ 
  while( $U \neq \emptyset$ )
  {
    for each  $x \in U$  {
      for each  $y \in S$  {
        if( $x$  dominates  $y$ )  $S = S - \{y\};$  next  $y$ ;
        elseif( $y$  dominates  $x$ )  $U = U - \{x\};$  next  $x$ ;
        else next  $y$ ;
      }
       $S = S \cup \{x\}; U = U - \{x\};$ 
    }
  }
}
```

At each point, the algorithm maintains the set S of survivors at this point, which have not been dominated by any of the alternatives considered so far. Each new, unconsidered, alternative C_i from U is now compared only with this set of survivors.

- If a current survivor dominates C_i , then C_i is eliminated without further compares. Since every alternative which is dominated by C_i will also be dominated by the survivor which dominated C_i , there is no loss involved in eliminating C_i from further consideration at this point.

- If any of the current survivors are dominated by C_i , they are removed from the stack.
- If upon reaching the end of set S , none of the present survivors dominate C_i , then C_i is added to S as a new survivor.

Given that only a small fraction of the total number of alternatives is expected to survive dominance, one would expect size of S at every iteration to be quite small, so that the overall complexity is expected to be much better than $O(n^2)$.

In order to compute the average-case time complexity of the algorithm, as noted by [12], the average number of compares at the p^{th} iteration is equal to the expected number of survivors at the p^{th} iteration, which is the same as $a(p, m)$, the expected number of survivors considering p alternatives and m criteria. Therefore the average-case time complexity of the overall algorithm can be expressed as,

$$\tau(n, m) = \sum_{p=1}^{n-1} a(p, m).$$

Or,

$$\tau(n, m) \leq \int_1^n a(p, m) dp.$$

(2.3)

so from Equation 2.2 $\tau(n, m) = O(n \ln^{m-1} n)$.

Kung [27] proposes an algorithm and derives the following time-complexities for the algorithm:

$$\tau(n, m) = \theta(n \ln n) \text{ for } m = 2, 3.$$

$$\tau(n, m) = O(n \ln^{m-2} n) \text{ for } m \geq 4.$$

2.3 Impact of correlated criteria

The performance analyses for the dominance filter presented so far are based on the assumption that the criteria are statistically independent or uncorrelated. In other words, it was assumed that no knowledge about the value taken by an alternative in one criterion can be gleaned by looking at any of its other criteria values. In

real world problems however, criteria are often correlated. We expect vehicles with higher city mileages to have higher highway mileages as well (positively correlated criteria). Conversely, vehicles with higher performance measures tend to score low on mileage efficiencies (negatively correlated criteria). A natural question is about the effect of such correlations on the performance measures, *filtering efficiency* and *time-complexity*, of the dominance filter. Since the time complexity of the described implementation of the Filter is closely tied to the filtering efficiency of the filter, the impact on the complexity can be inferred, and need not be discussed separately.

2.3.1 The Correlation Coefficient

Given two random variables, X_1 and X_2 , with means μ_1 and μ_2 , and standard deviations, σ_1 and σ_2 respectively, the *Correlation Coefficient* ρ_{12} between the two random variables is defined as:

$$\rho_{12} = \frac{E((X_1 - \mu_1) \cdot (X_2 - \mu_2))}{\sigma_1 \sigma_2}. \quad (2.4)$$

The term in the numerator is referred to as the *Covariance* of X_1 and X_2 . If C_1 and C_2 are column vectors of size n containing values on two criteria C_1 and C_2 of n alternatives, the Covariance between C_1 and C_2 is obtained as,

$$Cov(C_1, C_2) = \frac{1}{n} \sum_{k=1}^n (x_{k1} - \bar{x}_1)(x_{k2} - \bar{x}_2).$$

where \bar{x}_1 and \bar{x}_2 are the mean values for the criteria vectors C_1 and C_2 respectively. Now using Equation 2.4 the correlation coefficient between C_1 and C_2 can be written as:

$$\rho_{12} = \frac{1}{\sigma_1 \sigma_2} \left(\frac{1}{n} \sum_{k=1}^n (x_{k1} - \bar{x}_1) \cdot (x_{k2} - \bar{x}_2) \right). \quad (2.5)$$

It can be proved that $-1 \leq \rho_{xy} \leq 1$. If the correlation coefficient between two random variables is zero, the two variables are said to be uncorrelated. If $\rho_{xy} > 0$ then X and Y are said to be positively correlated and any increase in X is expected to increase Y . If $\rho_{xy} < 0$ then X and Y are said to be negatively correlated and an increase in

X is expected to bring about a decrease in Y . When $\rho_{xy} = 1$ or $\rho_{xy} = -1$, the two variables are said to be perfectly correlated, positively or negatively, respectively.

2.3.2 Filtering Efficiency and correlation coefficient

Suppose that C_i is one of n alternatives with criteria-vector $X = \{x_1, x_2, x_3, \dots, x_m\}$ such that ρ_{ij} is the correlation coefficient between criteria i and j respectively. In other words, the n criteria-vectors can be considered as drawn from the m -variate probability density function $f(X, \Sigma)$ where Σ is the covariance matrix containing the pairwise covariances between the criteria with the diagonal values containing the variances for the individual criteria, such that the individual ρ_{ij} s are maintained between the criteria. Then, the probability that another random alternative, C_j with evaluations $Y = \{y_1, y_2, y_3, \dots, y_m\}$ will dominate C_i is,

$$I(C_i) = \int_{x_m}^{\infty} \int_{x_{m-1}}^{\infty} \dots \int_{x_1}^{\infty} f(y_1, y_2, \dots, y_m) dy_1 dy_2 \dots dy_m \quad (2.6)$$

Then, $(1 - I(C_i))$ is the probability that C_i is not dominated by a random alternative from the remaining $(n - 1)$ alternatives. So, the probability that C_i is not dominated by any of the other $(n - 1)$ alternatives can be written as $(1 - I(C_i))^{n-1}$. Therefore, the probability that a random alternative C_i is a survivor from a set of n alternatives can be written as:

$$S(C_i) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} (1 - I(C_i))^{n-1} f(x_1, x_2, \dots, x_m) dx_1 dx_2 \dots dx_m \quad (2.7)$$

Provided the form of the distribution function, $f(X, \Sigma)$ is known, the expected number of survivors given the individual correlations between the criteria can be computed by multiplying the above quantity with n . However, even considering only two criteria, the above computation is quite complex as indicated by the above equations. Therefore, we conducted Monte-Carlo simulations to approximate the expected number of dominance survivors with varying values of n , and ρ for 2 criteria. The simulations consisted of generating a n points from a bivariate Gaussian distribution with correlation ρ and then applying the Dominance Filter to obtain the number of survivors.

Results obtained by Monte-Carlo simulations

Figures 3 and 4 respectively show the variation of the expected fraction and total number of dominance survivors with varying values of the correlation coefficient, ρ , between the two criteria. The plots were generated by means of Monte-Carlo trials conducted on normal vectors generated such that they are correlated by preselected values of ρ , and then applying the dominance filter to obtain the survivor sets ⁴.

The points for the plots are obtained over 100 trials conducted for the cases $n = 1000$, $n = 100$, $n = 10$ and $n = 2$. All cases are for 2 criteria. For the family of curves shown in Figures 3 and 4, three points on each of these curves can be obtained analytically, namely for values of ρ equal to -1.0 (all alternatives survive), $\rho = 0.0$ (as derived in [12] for independent criteria), and $\rho = 1.0$ (only one alternative survives) respectively.

The following additional observations can be made from these plots:

1. The experiments indicate that for a given n , the expected number of dominance survivors, as shown in figure 4, reduces as the value of ρ increases. This means that the result obtained for the expected number of survivors assuming independent criteria will be an overestimate when the criteria happen to be positively correlated, and an underestimate when the criteria are negatively correlated.
2. Figure 3 also suggests another interesting result seen in the region of very high negative correlation. When $\rho = -1.0$, we know that the fraction of survivors is 1.0. Now, as ρ is increased just a little bit so that the negative correlation is not perfect, we see that the fraction of survivors drops down drastically. For example, with $n = 1000$, Figure 4 shows that at $\rho = -0.9$, we have close to only 30 survivors, whereas at $\rho = -1.0$, we know that the expected number of survivors is 1000. This means that for $m = 2$, the filtering efficiency of the dominance filter is expected to deteriorate only in cases where the pairwise correlation happens to be negative and perfect. At all other values, the efficiency is expected to be close to that with independent criteria.

⁴Note that the minimum value of -1 on the Y-axis of the plot is an artifact of the plotting routine of the software used to generate these plots.

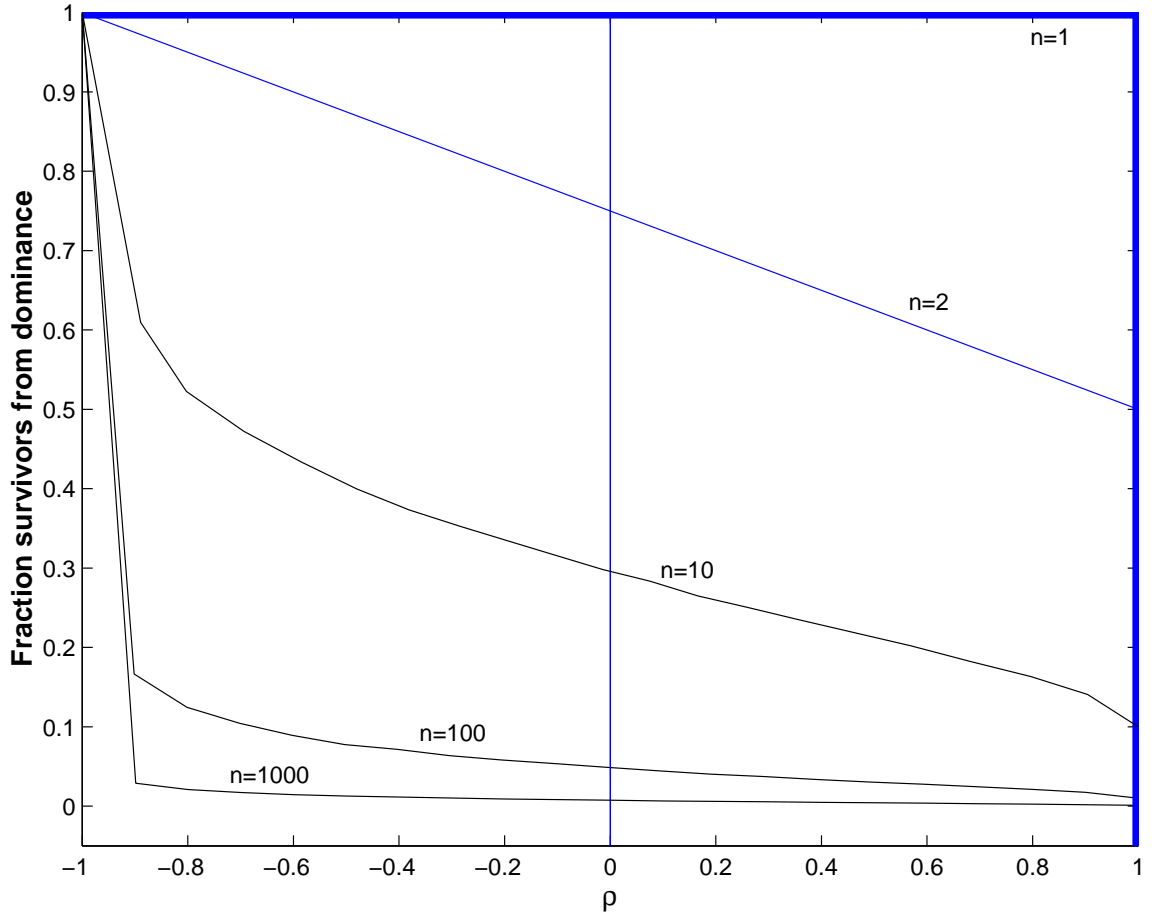


Figure 3: A plot of expected fraction of survivors for the dominance filter with varying values of the correlation coefficient.

- Figure 3 shows that for a given correlation, the expected fraction of survivors reduces as n increases. We have already seen this to be true with the case when the criteria are independent.

The above observations are true for a decision problem with $m = 2$. For the more general case with m criteria, the following observations apply:

- Even if a single pair of criteria from among the m criteria are perfectly, negatively correlated, all the alternatives will survive dominance irrespective of the evaluations along the remaining $m - 2$ criteria.

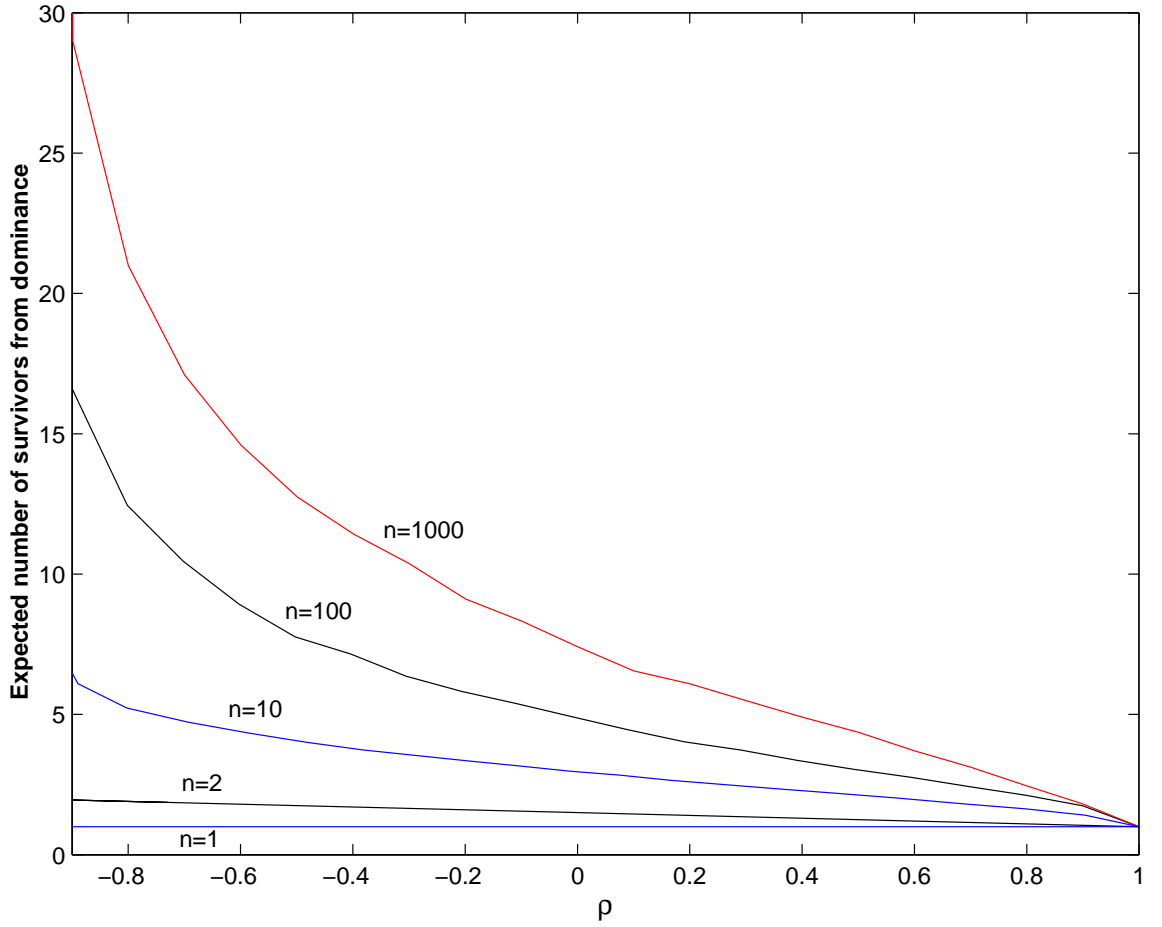


Figure 4: A plot of expected number of survivors for the dominance filter with varying values of the correlation coefficient.

2. The existence of a strong, imperfect negative correlation between a single pair need not necessarily deteriorate the filtering efficiency of the dominance filter since as shown in the previous section, even if a pair of criteria are strongly anti-correlated, their impact upon the filtering efficiency can be minimal. Hence, dominance filtering can be done even if such a pair of criteria are known to be present.
3. Theorem 2.1 shows that by adding new criteria to the existing set of m criteria, the expected number of survivors can only increase provided the alternatives

are distinct (no coincidence along all m criteria is allowed). The proof of this theorem contains no assumptions about independence among the m criteria, except for the requirement of *distinctness* which can be interpreted as requiring that at least one pair of criteria not be positively and perfectly correlated.

2.4 Summary

The dominance filter has many virtues with respect to filtering efficiency. The filtering is very effective for most problems except for problems with a large number of criteria and relatively small number of alternatives. The computational complexity of dominance is also $O(n \cdot \ln^{m-1} n)$ and it scales well with increasing number of alternatives. Experiments conducted to study the impact of correlated criteria on the performance of dominance filter show that for $m = 2$, unless the pair of criteria are almost perfectly anti-correlated, the filtering efficiency is comparable to the case for statistically independent criteria.

For the general case with m criteria, however, even if two criteria are perfectly, negatively correlated all alternatives survive the filter. Hence the presence of a pair of criteria that are perfectly, negatively correlated with each other can drastically reduce the filtering efficiency of the dominance filter. At the same time, if there is only one pair of criteria that are known to be anti-correlated, then as long as the correlation is not perfect, the filtering efficiency of the dominance filter is not affected as strongly. If however, more than a pair of criteria are anti-correlated, then the impact on the filtering efficiency of the dominance filter cannot be predicted based on the conducted experiments.

In the presence of a pair of strongly anti-correlated criteria, the Viewer can be used to express trade-offs along these two dimensions since they most require an expression of compromise from the user (a strong anti-correlation between two dimensions is also indicative of the extent to which it is difficult to maximize along both dimensions i.e., the need to express trade-off expressions along these dimensions). In the general case where the dominance filter is not efficient, the S-F-V architecture can still provide useful decision support by allowing the use of the Viewer to reduce the alternatives to a handful. The presence of negatively correlated criteria is just one kind of decision

situation for which the dominance filter will not maximally satisfy all the goals of effective decision support. In the next chapter, we will look at other such decision situations and motivate the need to extend the previously described S-F-V architecture so that it will satisfy most of the requirements for providing effective decision support.

CHAPTER 3

DECISION SITUATIONS AND CHOICE FILTERS

3.1 Decision Situations

A desirable property of DSS-architectures is that they be responsive both to the characteristics of the decision problem at hand, and to the related needs of the decision-maker. We will use the term *decision situation* to refer collectively to the problem characteristics and the user-needs for a decision problem. An individual user-need or problem characteristic will be referred to as a *situational demand*. An example of problem characteristics is uncertainty in the criteria values. In the presence of such uncertainty, the output of the dominance filter can no longer be assured to contain only Pareto-optimal alternatives. An example of user-needs is the user's stance towards the degree of pruning. A decision-maker might be more interested in being given a few good alternatives while another might want to make sure that he has many alternatives to work with.

In the S-F-V-architecture, the Filter component is designed to accommodate the abstract preferences of the decision-maker under various kinds of decision situations. In this chapter, we describe a few such decision situations which can be commonly associated with decision problems, and formulate corresponding alternatives to the dominance filter. The presence or absence of uncertainty in a decision problem directly limits the kinds of demands that a DM can impose. Therefore, in the dissertation, we broadly classify decision situations into two classes - those with uncertainty and those for which the criteria values are known accurately. We then consider the different

decision situations that arise due to the presence of various kinds of user-needs and other problem characteristics under each of these two classes. The goal is to be able to cater to all the decision situations implied by the above classification by deploying a suitable Filter in the S-F-V architecture. In the next section, we define *Choice Filters* and then formulate a particular family of dominance filters which can be deployed as choice filters in the S-F-V architecture to address various decision situations.

3.2 Choice Filters

A choice filter is a black box which takes as input the m criteria values for the n alternatives, the abstract preference expressions of the decision maker, along with situational demands, and produces a subset of the alternatives which will be referred to as the filter-survivors. Thus, a choice filter decides whether a multicriterially evaluated input alternative is to survive or be eliminated in a given decision situation

Ideally, the algorithm inside the box applies all of the decision-maker's preferences and at the same time is responsive to the decision situation. Constructing such a Filter is non-trivial because often the needs expressed by the user might be in conflict with some problem characteristics or other user-needs. For example, the decision problem might have a very large set of alternatives and the decision-maker might have indicated quite a large number of criteria as of interest to him, and that he desires a quick decision. Clearly, the time-complexity of processing a large dataset will need to be traded-off against the time of response desired by the decision-maker.

From the above definition of a choice filter, even an algorithm which selects the survivors randomly from the initial set qualifies as a choice filter although it might not satisfy a wide variety of situational demands. More practically, the decision science literature teems with techniques to solve the MCDM problem. Methods like elimination-by-aspects, goal programming, use of weights, marginal utilities and utility functions and other kinds of decision heuristics like lexicographic ordering also apply the decision-maker's preferences to produce a subset of the decision alternatives. Therefore, these techniques could also qualify as choice filters. The only problem with using any or most of the above as choice filters in the context of the S-F-V-architecture is their inability to address the different properties that a DSS must

possess, as discussed in Chapter 1, in order to elicit the decision-maker's preferences by means which are natural and easy for the decision-maker to express.

On the other hand, as discussed in Chapter 2, there is a very strong appeal to the use of dominance as a choice filter. But we have also seen examples wherein the use of the dominance filter might not be well-suited to catering to the related situational demands e.g, pruning demands requiring fewer than, or more than, the Pareto alternatives, accommodating uncertainty). Thus we are faced with the need to resolve the tension between the use of dominance as an integral part of any decision-making method and its inapplicability in dealing with certain situational demands. The question is whether it is possible to do this by making minor changes to the dominance rule itself so that all of its good properties are retained. Most of the choice filters formulated in the dissertation will be based on such decision rules- rules that can be derived by making minor adjustments to the dominance rule. First, we introduce some notations that will be used from here on to refer to choice filters and other related variables.

3.2.1 Terminology and Notations

A choice filter will be referred to as $F()$. Although $F()$ will be used for the general case, specific labels will be used to distinguish between different filters. The Dominance Filter presented in the last chapter will be represented as $D()$; it takes no parameters. A set of multicriterially evaluated alternatives will be represented as $\mathcal{X} = \{C, X\}$, where C is a set of labels, one for each of the n alternatives. C_k will be used to refer to the k th alternative in the set. X_i is an m -dimensional vector containing the m criteria values for the alternatives. In other words, for the alternative C_k ,

$$X_k = \{x_{k1}, x_{k2}, \dots x_{km}\}$$

so that the quantity x_{kj} will refer to the evaluation of the alternative C_k on the j th criterion. The set of alternatives that are produced upon application of a choice filter (the survivor set of the choice filter) will be represented by $S(F(), \mathcal{X})$. The first argument indicates the choice filter that is being applied while the second argument is the set of labeled alternatives to which the choice filter is being applied; the second

argument will often be omitted for convenience. The survivor set of a choice filter is a set of labels indicating the alternatives that survive the application of the filter. Therefore, $S(F(), \mathcal{X})$ is a set of labels. We will often need to refer to the criteria evaluations of the survivor set of a choice filter. The notation $X(S(F()))$ will be used to indicate the criteria values taken by the survivors of the filter $F()$.

For problems where there is uncertainty associated with the criteria values, we will use $\tilde{\mathcal{X}} = \{C, \tilde{X}\}$, where \tilde{X}_k is the set of m , *measured* criteria values for the alternative C_k . As before, $S(F(), \tilde{\mathcal{X}})$ will be used to indicate the survivors (labels for survivors) of the application of the filter F to the measured set of criteria values. Finally, $\tilde{X}(S(F()))$ refers to the measured criteria values of the survivors.

We say that two alternatives are ϵ -indistinguishable along some criterion if the values taken by the alternatives in that criterion are not separated by more than ϵ units. When we say that two alternatives are ϵ -indistinguishable without reference to a criterion, we mean that the alternatives are ϵ -indistinguishable along each of the m primary criteria.

We next consider a family of choice filters obtained by the introduction of tolerances in the dominance rule. This family of dominance filters will be referred to as *Tolerance-Based Dominance Filters* or TBDFs.

3.3 Tolerance-based Dominance Filters (TBDFs)

The choice filters presented in this section modify the dominance rule in various ways by the introduction of tolerances in the dominance rule. These tolerances are scalar values which can be thought of as either representing noise-accommodating thresholds or user-sensitivities to changes in criteria values. For each choice filter, we define the decision rule, the algorithm used, and discuss characteristics of the survivors produced by the algorithm. In the next section we consider the performance of these filters.

3.3.1 Single Pass Strict dominance tolerance filter- $ST(\epsilon)$

This choice filter produces its survivors after a single pass of the n alternatives. We will typically refer to this filter simply as *Strict dominance filter*.

Decision Rule:

$$(C_a \text{ beats } C_b \text{ by Strict Dominance}) \iff \exists i(C_{ai} > C_{bi} + \epsilon) \wedge \neg \exists j(C_{bj} > C_{aj} + \epsilon)$$

Algorithm:

The algorithm for this choice filter is the same as the one described for the Dominance Filter in Section 2.2.2. The only difference is that when a pairwise comparison is being made, the algorithm checks for the *Strict dominance* relation between the alternatives instead of the *dominance* relation.

Characterizing the survivors:

The *strictly dominates* relation, unlike the *dominates* relation, is not transitive. In other words, it is possible that an alternative C_i *strictly dominates* alternative C_j , C_j *strictly dominates* alternative C_k and for C_i to not *strictly-dominate* alternative C_k . It is possible that either C_i and C_k tie or even that C_k *strictly dominates* C_i . In the former case, the *strictly dominates* relation is being *weakly intransitive*⁵. In the latter case, we say the *strictly dominates* relation is being *strongly intransitive*⁶.

To see an example of the weak intransitivity of strict dominance, consider the 3 alternative case with 2 criteria p and q as indicated below-

	p	q	
C_i	2	3	
C_j	3	1	(e1)
C_k	1	2	

Let $\epsilon = 1$ for both criteria p and q. Then, for the above case we see that C_i *strictly dominates* C_j , C_j *strictly dominates* C_k , but C_i doesn't *strictly dominate* C_k . Rather C_i and C_k are tied under the *strictly dominates* relation. The triplet (C_i, C_j, C_k) in the example is referred to as an *intransitive chain*.

⁵A relation R is said to be weakly intransitive if for some a,b, and c, we have: $(aRb) \wedge (bRc) \wedge \neg(aRc)$

⁶A relation R is said to be strongly intransitive if for some a,b, and c, we have: $(aRb) \wedge (bRc) \wedge (cRa)$

To see that the *strictly dominates* relation can be strongly intransitive, consider the 3 alternative, 3 criteria example again with ϵ values all set to 1.

	p	q	r	
C_i	1	3	2	
C_j	2	1	3	(e2)
C_k	3	2	1	

In the above case we see that C_i strictly dominates C_j , C_j strictly dominates C_k and C_k strictly dominates C_i . The triplet (C_i, C_j, C_k) is referred to as an intransitive loop. Theorem 3.1 below shows that strong intransitivity cannot occur for a case with less than 3 criteria.

Theorem 3.1 *The Strict Dominance rule cannot be strongly intransitive for a decision problem with $m < 3$.*

Proof: See Appendix A.

The result of these breakdowns in transitivity is that the single pass algorithm for the previous dominance filter is now seen to be order-dependent. To see this, consider the previous example assuming that we use the single pass algorithm described. Suppose that the algorithm starts with the pair C_i, C_j . Since C_i strictly dominates C_j in a pairwise comparison, the algorithm will discard C_j and go to the next unconsidered alternative C_k . Upon finding that C_k strictly dominates the survivor C_i , the algorithm discards C_i so that the only survivor produced by the algorithm is C_k . But clearly, there is a case to be made for why C_i and C_j deserve to be survivors- C_j strictly dominates C_k and C_i strictly dominates C_j . Also, all three alternatives are Pareto-optimal. But the single-pass algorithm will always discard two of them due to order-dependency.

There are at least two ways to tackle the problem associated with order-dependence of the single-pass algorithm. Choice filters based on each of these techniques make use of the Strict dominance rule above, but operate using more than a single pass in order to produce the set of survivors. These two filters, Twopass Strict dominance filter, and Multipass Strict dominance filter, will be discussed in the next section as possible alternatives to tackle the order-dependence of the single-pass algorithm.

We already showed above that the strict dominance filter can eliminate Pareto optimal alternatives. It can additionally be shown that the filter can retain suboptimal alternatives as survivors. To see this, consider the 3 alternatives,

	x	y	
C_i	3	1	
C_j	1	2	(e3)
C_k	2.5	0.5	

Let the ϵ s for both the criteria be 1. Then, C_i and C_k survive while C_j is eliminated. However, C_k is not Pareto-optimal while the eliminated alternative C_j is optimal.

3.3.2 Twopass Strict dominance tolerance filter- $ST2(\epsilon)$

In the discussion of the single pass strict dominance filter, it was shown how a single pass algorithm which applies the strict dominance rule can result in a set of survivors which are dependent upon the order in which the alternatives are compared. For many applications, losing alternatives due to breakdown in transitivity can be undesirable for the decision-maker, especially if the alternatives are Pareto optimal. Also, the set of survivors produced by the application of a non-transitive decision rule is difficult to characterize for the decision-maker. The twopass strict dominance filter is one alternative to tackle the order-dependence of the original strict dominance filter.

To understand the rationale behind this filter, suppose that the single pass strict filter retains a few alternatives that belong to intransitive loops. During its second pass, the Twopass Strict filter checks if any of the survivors of the single pass are *strictly dominated* by a non-survivor. Since for every loop representative retained by the first pass, there will be one such non-survivor, all such loop representatives will be removed during the second pass. In the case of an intransitive chain, if the single pass keeps the single alternative at the tail of the chain (the one which is strictly-tied with the head of the chain), then the second pass eliminates such alternatives). In other words, at the end of the second pass, none of the survivors are *strictly dominated* by a non-survivor. Hence the twopass strict filter produces the set of *strictly undominated* alternatives.

Decision Rule:

$$C_i \in ST2(\epsilon) \iff \neg \exists C_j (C_j \text{ strictly dominates } C_i)$$

Algorithm:

1. Produce the survivors of the order-dependent single pass strict algorithm.
2. Eliminate any survivor that is strictly dominated by a non-survivor.

Characterizing the survivors:

In other words, the Twopass Strict dominance filter produces a survivor set with the following property: 'no survivor is strictly dominated by any of the other alternatives.' The algorithm described above runs in two passes. The first pass is the single pass strict dominance filter and this produces a set of survivors for which the following are true:

- No survivor strictly dominates another survivor and,
- Some survivors may be *strictly dominated* by non survivors of the first pass.

The second pass of this algorithm gets rid of the survivors which are strictly dominated by the non-survivors. Hence the second pass iterates over the set of survivors asking for each survivor if it is dominated by any of the non surviving alternatives. If so, then the survivor at hand is eliminated and process iterates over the remaining survivors. Not more than two passes are needed to produce a strictly-undominated set of alternatives. Hence in the worst case, the amount of work done by this implementation is twice the amount of work done by its predecessor. Thus, the complexity remains unchanged although it takes a little longer to produce its set of survivors. Also, the set of survivors produced by this filter is a subset of the set of survivors produced by the single pass filter. Hence this filter has a better pruning efficiency than the single pass filter.

One problem with this filter is that it can potentially produce zero survivors as can be seen for the example (e2) discussed in the previous section. In the example,

each of C_i , C_j and C_k are strictly dominated by another alternative and hence this scheme will produce zero survivors. But it is fairly trivial to have a post-processor in the implementation which checks for this condition and if the second pass leads to zero survivors, it reverts to some other decision rule to produce an alternative non-empty set of survivors.

3.3.3 Multipass Strict dominance tolerance filter– $STm(\epsilon)$

The previous section described one way to tackle the transitivity problem associated with the single pass Strict dominance filter. Another way to solve this problem is to allow for each and every alternative occurring in a loop or chain to survive, so long as it is not strictly dominated by another alternative except by those which are also in the loop or chain. Clearly, this filter will produce more survivors than that produced by the single pass, or the twopass strict dominance filters.

Decision Rule:

$$C_i \in STm(\epsilon) \iff \forall C_j ((C_j \text{ strictly dominates } C_i) \rightarrow (C_j \text{ occurs in an intransitive loop or chain with } C_i))$$

In order to implement this filter, we need to seek out all those alternatives which occur in loops and chains but which are not strictly dominated by any non-loop or non-chain alternatives, i.e., alternatives like C_i and C_j in the example (e2). This algorithm is a bit complicated and is therefore described below more formally. If there are loops and chains, then it will take more than 2 passes to produce the final set of survivors.

Algorithm:

The first pass is obviously the single pass strict dominance filter as before. The subsequent passes work as follows:

Second Pass: Iterates over the non-survivors of the first pass, looking for any such non-survivors which are not strictly-beaten by any of the first-pass alternatives. This pass will fetch all such alternatives which occur at the end of a chain.⁷

Third pass: Iterates over the existing set of non-survivors now looking for all such non-survivors which strictly dominate an existing survivor. If so, then such non-survivors are salvaged as new survivors but are stored separately.

Fourth pass: Performs the same check on the existing set of non-survivors as the third pass except that the check is performed only with respect to the newly salvaged survivors of the third pass. The new survivors brought in by the fourth pass are again stored separately.

Remaining passes: Each pass from here on, works exactly like the fourth pass, looking for existing non-survivors which strictly dominate any of the survivors which were salvaged by the previous pass. This is continued until none of the remaining non-survivors can be salvaged as a survivor any more.

The third pass and the passes thereafter fetch alternatives which occur in a loop or alternatives that occur in the middle of a chain. By storing the newly salvaged survivors of each of these passes separately, the time taken by each of the passes is reduced considerably. This is important since it is sufficient to keep comparing only with respect to the survivors of the previous pass in order to produce the entire set of alternatives that occur in a loop or a chain. In other words, each pass i salvages one non-surviving alternative, say X_i , from each loop and the $(i + 1)$ -st pass gets the alternative, X_j , which strictly dominates X_i in the loop, and the $(i + 2)$ -nd alternative fetches the alternative which strictly dominates X_j and so on until all the alternatives in each loop are salvaged. The total number of passes required by this algorithm is equal to the size of the largest undominated loop, s , encountered by the first pass. However, after the third pass, each subsequent pass does not have to iterate over all of the existing survivors due to reasons discussed above. Each pass iterates over all of the existing non-survivors until the strict-dominators for each of the previous-pass survivors are found, which requires less than n compares. The application of this filter

⁷Alternatives occurring at the end of a chain are by definition strictly-tied with the alternative which occurs at the head of the chain and we know that the head of a chain always survives the first pass.

can therefore produce undesirable delays if n is large and under conditions wherein the presence of big loops become probable.

Suppose that $\frac{n}{k_i}$ is the number of survivors produced by the i^{th} pass. Then, the $(i+1)st$ pass maximally requires comparing each of the remaining non-survivors to the $\frac{n}{k_i}$ survivors of the i^{th} pass. Hence, for s being the size of the largest loop, the total number of pairwise comparisons required by the Multipass Strict filter excluding the first pass is:

$$\tau(n, m) = \sum_{j=2}^s \left(n - \sum_{i=1}^{j-1} \frac{n}{k_i} \right) \cdot \frac{n}{k_{j-1}} \quad (3.1)$$

which, including the first pass (Strict dominance), is $O(n^2)$ in the worst-case.

3.3.4 Strongly-Strict dominance filter - $Sts(\epsilon)$

The Strongly-Strict filter is another filter that belongs to the original Strict dominance family of filters.

Decision Rule:

$$C_i \in Sts(\epsilon) \iff (\forall C_j \in C) \exists k (\tilde{x}_{ik} > \tilde{x}_{jk} + \epsilon).$$

where $j \neq i$.

Clearly the above rule is neither transitive, nor even antisymmetric (meaning two alternatives can *beat* each other by this rule). However, the application of this filter by the use of an order-independent algorithm produces a survivor set that has interesting properties.

The Algorithm:

1. Produce dominance survivors by applying the dominance rule to the measured criteria values.
2. To the dominance survivors, apply the single pass Strict filter.

3. Obtain the set of strictly undominated alternatives from the survivors of the second pass.
4. From this set of survivors, further eliminate those which have at least one alternative in any of its ϵ -boxes.

The time-complexity of the above algorithm is also $O(n \cdot \ln^{m-1} n)$ like the other TBDFs. To see this, note that the each pass subsequent to the first one maximally compares all of the dominance survivors to all of the dominance non-survivors.

In order to see that this algorithm satisfies the decision rule, we need to prove that all survivors of the algorithm satisfy the decision rule and that none of the non-survivors satisfy the same rule. Let us consider the first part - that of showing that all survivors of the above algorithm satisfy the decision rule.

At the end of the algorithm above, we have a set of alternatives none of which are strictly dominated by any of the remaining $n - 1$ alternatives. This implies that each survivor either strictly dominates another alternative or it is *strictly tied* with another alternative (two alternatives are said to be strictly tied if neither dominates the other according to the strict dominance rule). Now strict-ties can be classified into two kinds - *weak ties* wherein neither of the two alternative strictly dominates the other because they are ϵ -indistinguishable with each other along all criteria or *strong ties* wherein, the two alternatives are each ϵ -better than the other along different criteria. Step 4 of the algorithm however eliminates all such survivors that are *weakly-tied* with some alternative. Therefore, we can now say the following: each survivor of the algorithm are such that they either strictly dominate another alternative or they from strong, strict-ties with another alternative. This implies that each survivor of the algorithm satisfies the decision rule.

To prove that no non-survivor of the above algorithm satisfies the decision rule, we first note that none of the non-survivors of the first step can satisfy the decision rule (since they are dominance non-survivors, they cannot be epsilon-better on some criterion with respect to their dominator(s)). Also, step 2 eliminates those survivors of step 1 which are strictly-dominated by another survivor, hence these new non-survivors cannot satisfy the decision rule either. Step 3 eliminates alternatives that are strictly dominated by any alternative and hence removes alternatives that

cannot possibly satisfy the decision rule. Finally, if there are survivors of Step 3 that are dominance-tied but ϵ -indistinguishable along all criteria, Step 4 eliminates such alternatives. This removal is also valid since ϵ -indistinguishable alternatives violate the the decision rule with respect to each other. In summary, none of the non-survivors satisfy the decision rule for the strongly-strict filter. This shows that the above algorithm produces the set of strongly-strict non survivors.

Characterizing the survivors:

The name of the filter is based on the fact that *Strict-ties* (namely pairs of alternatives where neither *strictly dominates* the other) which previously resulted from ϵ -indistinguishability are no longer considered as valid ties; only *ties* of the kind where each of the two alternatives are ϵ -better than the other alternative on different criteria are considered to be valid ties. Hence one way to characterize the survivors of the Strongly Strict filter is that it is the set of survivors produced by the Twopass Strict filter with all weakly-tied survivors being eliminated.

The following two properties can be proved for the survivors produced by the Strongly Strict filter:

Corollary 3.1 *Every survivor of the Strongly Strict filter is a dominance survivor, i.e.,*

$$S(Sts(\epsilon), \tilde{\mathcal{X}}) \in S(D, \tilde{\mathcal{X}})$$

Proof: See Appendix A.

Theorem 3.2 *For $\tilde{\mathcal{X}} = \{C, \tilde{\mathcal{X}}\}$,*

$$\emptyset = S(Sts(\epsilon_k), \tilde{\mathcal{X}}) \subseteq \cdots S(Sts(\epsilon_3), \tilde{\mathcal{X}}) \subseteq S(Sts(\epsilon_2), \tilde{\mathcal{X}}) \subseteq S(Sts(\epsilon_1), \tilde{\mathcal{X}}) = S(D, \tilde{\mathcal{X}})$$

where,

$$\epsilon_k > \cdots > \epsilon_2 > \epsilon_1 \geq 0.$$

In other words, the Strongly Strict filter allows for a smooth pruning of the Pareto set with increasing ϵ . As stated earlier, the survivor set produced by the original Strict dominance filter can contain alternatives that are not in the Pareto set. Also, the size of the survivor set is not a monotonic function of the magnitude of the tolerance. As we will see in Chapter 6, there are decision situations where it becomes necessary

to be able to produce gradually reducing, and exclusive, subsets of the dominance survivor set by changing the magnitude of the tolerance. The Strongly Strict filter suits perfectly for this requirement as indicated by the above two results.

3.3.5 Superstrict dominance tolerance filter- $SS(\epsilon)$

Decision Rule:

$$(C_x \text{ beats } C_y \text{ by Superstrict Dominance}) \iff \exists i(C_{xi} > C_{yi} + \epsilon) \wedge \forall j(C_{xj} \geq C_{yj} + \epsilon)$$

Characterizing the survivors:

Corollary 3.3, to be stated soon, shows that all dominance survivors are also survivors of the Superstrict dominance rule. Theorem 3.3 below shows that progressively increasing the size of the tolerances produces progressively more inclusive survivor sets.

Theorem 3.3 For $\tilde{\mathcal{X}} = \{C, \tilde{X}\}$,

$$\begin{aligned} S(D, \tilde{\mathcal{X}}) &= S(SS(\epsilon_1), \tilde{\mathcal{X}}) \subseteq S(SS(\epsilon_2), \tilde{\mathcal{X}}) \subseteq S(SS(\epsilon_3), \tilde{\mathcal{X}}) \\ &\dots \subseteq S(SS(\epsilon_k), \tilde{\mathcal{X}}) = C. \end{aligned}$$

where,

$$0 \leq \epsilon_1 < \epsilon_2 < \dots < \epsilon_k$$

Hence for the Superstrict dominance filter, as the tolerance increases, more and more alternatives are added to the dominance set. In other words, the survivors of the Superstrict dominance filter can be characterized as consisting of all of the dominance survivors and potentially some of the dominance non-survivors (the size of this additional set of survivors is a non-decreasing function of the size of the tolerance). This theorem also indicates that the Superstrict filter acts as a dual to the Strongly strict filter with respect to the Pareto set. This can be observed by comparing the above theorem to Theorem 3.2. In fact, the two theorems allow us to infer the following

interesting result: For $\tilde{\mathcal{X}} = \{C, \tilde{X}\}$

$$\begin{aligned}
\emptyset &= S(Sts(\alpha_1)) \subseteq S(Sts(\alpha_2)) \subseteq S(Sts(\alpha_3)) \cdots \\
&\subseteq S(Sts(\alpha_p)) = S(D) = S(SS(\omega_q)) \\
&\subseteq S(SS(\omega_5)) \subseteq S(SS(\omega_6)) \cdots \subseteq S(SS(\omega_r)) = C.
\end{aligned} \tag{3.2}$$

where $\alpha_1 > \alpha_2 > \alpha_3 \cdots > \alpha_p \geq 0 \leq \omega_q < \omega_5 < \omega_6 \cdots < \omega_r$.

Also, we see from the decision rule above that if alternative C_i *superstrictly dominates* alternative C_j , then it implies that C_i *Pareto dominates* C_j as well. In other words a dominance non-survivor can be *superstrictly dominated* only by its Pareto dominators. In other words, each dominance non-survivor which is retained by the Superstrict filter is such that none of its Pareto-dominators *superstrictly dominates* it. This motivates an algorithm to produce the survivors of Superstrict dominance consisting of two passes with the first pass producing the dominance survivors and the second pass additionally retaining each such non-survivor of the first pass, for which none of the first pass survivors *superstrictly dominates* it.

Algorithm:

1. Apply the dominance rule to produce the set of dominance survivors.
2. For each non-survivor of the first pass, check if any of the first-pass survivors superstrictly dominates it. If not, retain such alternatives as new survivors.

By using the above twopass algorithm instead of the original single pass algorithm used for the Dominance filter, we make the time complexity of producing the Superstrict survivor set independent of the additional alternatives produced by Superstrict over dominance. The complexity of the first pass is same as that of producing the dominance set and the second pass requires comparing each non-survivor once with each dominance survivor and this is of the order of the same number of pairwise compares as the dominance filter.

3.3.6 DD-dominance tolerance filter- $DD(\epsilon)$

Decision Rule:

$$(C_x \text{ beats } C_y \text{ by } DD - \text{Dominance}) \iff \exists i(C_{xi} > C_{yi} + \epsilon) \wedge \forall j(C_{xj} \geq C_{yj})$$

Characterizing the survivors:

The dd-dominance rule was originally conceived to modify the Strict dominance rule so that the transitivity problems associated with it can be addressed. The rationale of the above modification can be elicited by looking at the example (e2) again which contains the loop (C_i, C_j, C_k) . Previously it was stated that C_i *strictly dominates* C_j *strictly dominates* C_k *strictly dominates* C_i . However, careful observation will reveal that the criterion on which each of these alternatives beat their neighbor is different. C_i beats C_j on criterion y, C_j beats C_k on criterion z and C_k beats C_i on criterion x. Since each pair is therefore being effectively compared only on mutually exclusive dimensions (i.e. they are indistinguishable on the rest), clearly there is no reason to expect transitivity. This is exactly the phenomenon that manifests itself in example (e2) resulting in the intransitivity of the *strictly dominates* relation.

If it is ensured that either all criteria or at least 1 criterion plays an effective part in every pairwise comparison, then weak transitivity has to follow (no loops). The relation dd-dominates does the former - it makes sure that criteria which get ruled out of the pairwise comparison due to indistinguishability necessarily play a part in the fate of the pairwise comparison. Now even if C_i is ϵ better than C_j on y, it is really worse than C_j on criteria x and z. Therefore C_i no longer beats C_j so that both C_i and C_j , and by similar argument C_k , all survive. This ensures transitivity. Since the second clause of the dd-dominates relation requires that the dominator be strictly better than the dominated alternative, the dd-dominates relation entails the dominates relation. In other words, if C_i dd-dominates C_j then C_i dominates C_j . This is better stated as the Theorem 3.1 below,

Theorem 3.4 *Every survivor of the dominance filter is a survivor of the dd-dominance filter, i.e.,*

$$S(D, \tilde{\mathcal{X}}) \subseteq S(DD(\epsilon), \tilde{\mathcal{X}})$$

Proof: See Appendix A

According to Theorem 3.4, dd-dominance keeps all the dominance survivors. Similar to the Superstrict filter, the dd-dominance filter can potentially retain more, with the number being directly proportional to the size of the tolerances used. Just like the Superstrict filter, any dominance non-survivor can be dd-dominated only by its dominator. Hence, the algorithm to produce the survivors of the dd-dominance filter is similar to that for the Superstrict filter.

Algorithm:

1. Apply the dominance rule to produce the set of dominance survivors.
2. For each non-survivor of the first pass, check if any of the first-pass survivors dd-dominates it. If not, retain such alternatives as new survivors.

It is easy to see that each second pass survivor is such that it is within ϵ on all criteria with respect to all of its dominators.

Theorem 3.5 shows the relation between the survivor sets produced by the dd-dominance filter and the twopass strict dominance filter.

Theorem 3.5 *Every survivor of Twopass Strict Dominance is a survivor of dd dominance for the same tolerance value i.e.,*

$$S(ST2(\epsilon), \tilde{\mathcal{X}}) \subseteq S(DD(\epsilon), \tilde{\mathcal{X}})$$

Proof: See Appendix A

3.3.7 ϵ -Box Twopass Filter (ebtp-filter)- $EB(\epsilon)$

Decision Rule:

$$C_i \in EB(\epsilon) \iff (C_i \text{ is a dominance survivor}) \cup (C_i \text{ is } \epsilon - \text{indistinguishable from a dominance survivor})$$

Algorithm:

1. Apply the dominance filter to produce dominance survivors.
2. For each non-survivor of the first pass, check if it is within ϵ on every criteria from a first-pass survivor. If so, such alternatives are further added to the survivor set.

Characterizing the survivors:

The set of survivors of the ebtp-filter has the property P that every survivor is either a dominance survivor or it is inside the ϵ -hyperbox of some dominance survivor. From the discussion of the dd-dominance filter, we know that each second pass survivor of the dd-dominance filter is such that it is inside the ϵ -hyperbox of *all* of its dominators. Since the first pass is identical for both the filters, it is easy to see that all survivors of dd-dominance will also survive the ebtp-filter. Conversely, a second pass survivor of the ebtp-filter need not be in the ϵ -box of *all* of its dominators. Hence, the ebtp-filter can potentially produce more survivors in addition to the survivors produced by the dd-dominance filter.

Some theorems and corollaries establishing the relations between the survivor sets produced by the different choice filters described so far are stated below.

Theorem 3.6 *Every survivor of the ebtp-filter is a survivor of the Superstrict dominance filter for the same tolerance value i.e.,*

$$S(EB(\epsilon), \tilde{\mathcal{X}}) \subseteq S(SS(\epsilon), \tilde{\mathcal{X}})$$

Proof: See Appendix A

Theorem 3.7 *A survivor of the Superstrict dominance filter need not be a survivor of the ebtp-filter for the same tolerance value.*

Proof: See Appendix A

This implies that Superstrict dominance will produce a larger number of survivors than the ebtp-filter. Not only does the Superstrict filter get all the survivors of the ebtp-filter but it can potentially retain more alternatives as survivors. In this sense the Superstrict filter is more *conservative* than the ebtp-filter.

Theorem 3.8 *Every survivor of dd-dominance filter is a survivor of the ebtP-filter for the same tolerance value i.e.,*

$$S(DD(\epsilon), \tilde{\mathcal{X}}) \subseteq S(EB(\epsilon), \tilde{\mathcal{X}})$$

Proof: See Appendix A

Theorem 3.9 *A survivor of the ebtP-filter need not be a survivor of the dd-dominance filter for the same tolerance value.*

Proof: See Appendix A

Corollary 3.2 $S(ST2(\epsilon), \tilde{\mathcal{X}}) \subseteq S(DD(\epsilon), \tilde{\mathcal{X}}) \subseteq S(EB(\epsilon), \tilde{\mathcal{X}}) \subseteq S(SS(\epsilon), \tilde{\mathcal{X}})$

Proof: This follows from Theorems 3.5, 3.6, and 3.8

Corollary 3.3 *Every dominance survivor is a survivor of Superstrict dominance filter, i.e.,*

$$S(D, \tilde{\mathcal{X}}) \subseteq S(SS(\epsilon), \tilde{\mathcal{X}})$$

Proof: This follows from Theorem 3.4 and Corollary 3.2

3.3.8 Onion-skin filter- $OS(r)$

Decision Rule:

$$C_i \in OS(r) \iff \bigcup_{k=1}^r (C_i \in S(D, (\mathcal{X} - \bigcup_{j=1}^k \mathcal{D}_{j-1})))$$

where $\mathcal{D}_p = S(D, (\mathcal{X} - \mathcal{D}_{p-1}))$, $\mathcal{D}_0 = \emptyset$.

Algorithm:

```

Peelskin(r){
  O = ∅; U = X;
  while(r ≠ 0){
    O = O ∪ S(D, U);
  }
}

```

$$\begin{aligned}
&U = U - S(D, U); \\
&r = r - 1; \\
&\} \\
&\}
\end{aligned}$$

Characterizing the survivors:

$OS(r)$ is the classical dominance filter applied iteratively r times to the eliminated alternatives at hand. This process can be visualized as very similar to the peeling of an onion-skin. The top skin represents the Pareto set. Upon removal of this set, a new top skin is visible which can itself be peeled by the application of dominance filter to the non-survivors to retrieve yet another layer of Pareto survivors. These layers can each be named $\{PO\}_1$, $\{PO\}_2$ and so on, where $\{PO\}_i$ is the set of alternatives obtained after i applications of the dominance filter to the existing set of non-survivors.

Large chunks of increase in the number of survivors over that produced by the dominance filter can be obtained by the use of this filter. If n is large, the fraction of survivors upon application of a single round of measured dominance is quite small. As a result, the second application will produce almost the same number of survivors as the first one and so on for quite a few number of initial applications of measured dominance. Thus user expressions like "produce k times the number of measured Pareto dominance survivors" can be met by peeling k skins from the original set by the use of the above filter. Thus, the onion-skin filter can address some situational demands both in the presence or in the absence of uncertainty. If the desired increase in the number of survivors over the dominance survivors is not in large chunks, then a stronger notion of *neighborhood* will need to be applied instead of the onion-skin filter.

In the next section, we analyze the performance characteristics of some of the choice filters discussed above.

3.4 Performance characteristics of choice filters

The performance characteristics of choice filters provide us with knowledge about how the filters behave with respect to varying kinds of problem characteristics and user-needs.

We have already looked at how the pruning efficiency and time-complexity of the dominance filter varies with respect to various problem characteristics like n , m and correlations among criteria in Chapter 2. We next look at these performance characteristics for the choice filters discussed in the previous section.

3.4.1 Filtering efficiency

The filtering efficiency for the dominance filter is expressed as a function of n and m . For TBDFs, we want an expression in terms of n , m , and ϵ . In order to find an expression for the probability that a random alternative X is a survivor with respect to $n-1$ randomly chosen alternatives for a certain filter, it is useful to refer to Figure 5. The figure shows a simple case with 2 criteria for each of 4 filters; X is some arbitrary alternative shown as a point in a plot with the criteria as the axes. The coordinates of X , x_1 and x_2 , therefore correspond to the criteria evaluations for X respectively in the two criteria. The box around the point X is a square of side 2ϵ , with X at its center. This geometric representation will allow us to write an expression for the probability that X survives $n-1$ other, randomly chosen alternatives. Let's assume that the criteria values are obtained from a continuous, uniform distribution in the interval $[0,1]$. All tolerance values therefore lie in the same interval. Figure 5a shows the case for the Dominance filter discussed in the last chapter. The region Q seen as formed to the top and right of the solid lines emanating from X is a sub-region in the C1-C2 region such that any alternative that lies in Q will be a dominator of X . This geometric relation is inferred by using the dominance rule and observing from the figure that any alternative in the region Q will be such that it will dominate X according to the rule. Therefore the survival probability of X is indirectly proportional to the area of the region Q . Since the overall area of the C1-C2 plot region is 1.0, the area Q can be computed as $1.0 - (\text{total area of the non-}Q \text{ regions of the plot})$.

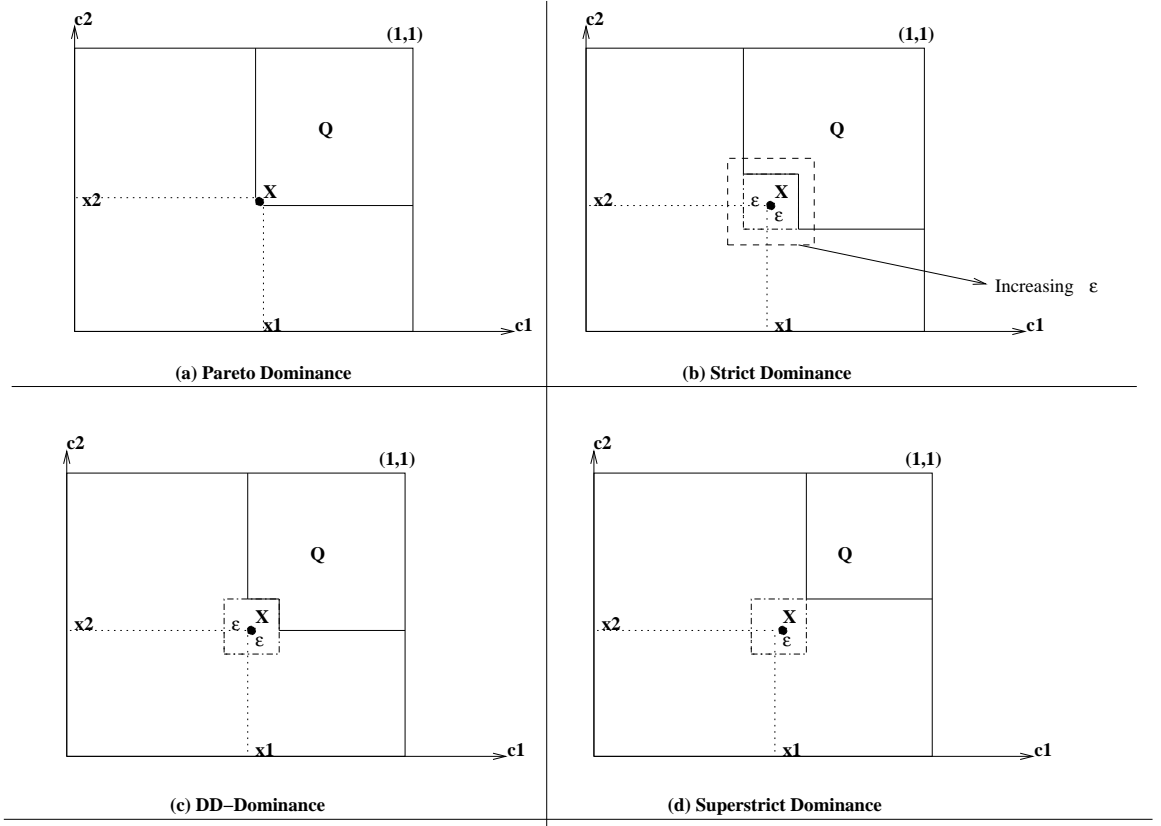


Figure 5: A geometric representation for a 2 criteria problem to compute the survival probability of a random alternative X for various TBDFs.

The same geometric analogy is true for the TBDFs as well except that the region Q now also depends upon the value of the tolerance, ϵ . Each of the other three figures show the region Q and the corresponding solid lines that bound Q from the lower end. From the figures it is seen that for a chosen X , the area Q is the smallest for the Superstrict Filter compared to the DD-dominance filter, which is smaller than the Q region for the Strict filter. Hence, the survival probability of an arbitrary X will be higher when the Superstrict dominance rule is applied than when the DD-dominance rule is applied which is further greater than the survival probability for the Strict Filter.

A closer examination of the plots will also show that except for the Strict filter, the area of the region Q reduces monotonically as the magnitude of the tolerance

increases. To do this, one can imagine another, larger box around point X and look at the equivalent Q region produced by this new box. This implies that the number of survivors produced both by DD-dominance filter and Superstrict filter will increase as the tolerance values increase.

For the Strict filter, the change in the area of region Q cannot be predicted by the direction of change in the tolerance values. This is because the new Q region produced by a larger box (shown as dotted box in Figure 5b) for the Strict filter will include previously excluded regions and also exclude some of the region which were previously in the Q region. So the net change in the area of the region Q cannot be predicted. This means that we cannot in general expect the number of survivors to either increase or decrease based merely upon the direction of change in the tolerance values. Expressions for the probabilities that an arbitrarily chosen alternative $X = \{x_1, x_2, \dots, x_m\}$ will survive each of the above TBDFs, given n, m, ϵ 's, and assuming that the criteria values come from a continuous uniform distribution in $[0,1]$ are given below. The expressions are based on computing the area of the individual Q regions and taking its complement by subtracting from 1.0 and integrating over all possible criteria evaluations that X can take. The expressions are generalizations of the 2 criteria case described above but the geometric ideas still apply except that the Q regions now become m-dimensional volumes.

- Strict Dominance:

$$P(X \in S(St(n, m, \epsilon))) = \underbrace{\int_0^1 \int_0^1 \dots \int_0^1}_{m \text{ integrals}} [1 - f(x_1)f(x_2)\dots f(x_m) + g(x_1)g(x_2)\dots g(x_m)]^{n-1} dx_1 dx_2 \dots dx_m$$

where,

$$f(x) = \begin{cases} 1 & 0 \leq x \leq \epsilon \\ 1 - x + \epsilon & \epsilon \leq x \leq 1 \end{cases}$$

and for $\epsilon \leq 0.5$,

$$g(x) = \begin{cases} x + \epsilon & 0 \leq x \leq 1 - \epsilon \\ 2\epsilon & \epsilon \leq x \leq 1 - \epsilon \\ 1 - x + \epsilon & 1 - \epsilon \leq x \leq 1 \end{cases}$$

for $\epsilon \geq 0.5$,

$$g(x) = \begin{cases} x + \epsilon & 0 \leq x \leq 1 - \epsilon \\ 1 & 1 - \epsilon \leq x \leq \epsilon \\ 1 - x + \epsilon & \epsilon \leq x \leq 1 \end{cases}$$

- DD-Dominance:

$$P(X \in S(DD(n, m, \epsilon))) = \underbrace{\int_0^1 \dots \int_0^1}_{m \text{ integrals}} [1 - (1 - x_1)(1 - x_2) \dots (1 - x_m) + g(x_1)g(x_2) \dots g(x_m)]^{n-1} dx_1 \dots dx_m$$

where,

$$g(x) = \begin{cases} \epsilon & 0 \leq x \leq 1 - \epsilon \\ 1 - x & 1 - \epsilon \leq x \leq 1 \end{cases}$$

- Superstrict-Dominance:

$$P(X \in S(SS(n, m, \epsilon))) = \underbrace{\int_0^1 \int_0^1 \dots \int_0^1}_{m \text{ integrals}} [1 - f(x_1)f(x_2) \dots f(x_m)]^{n-1} dx_1 dx_2 \dots dx_m$$

where,

$$f(x) = \begin{cases} 1 - x - \epsilon & 0 \leq x \leq 1 - \epsilon \\ 0 & 1 - \epsilon \leq x \leq 1 \end{cases}$$

The above expressions can be used to compute the expected number of survivors by multiplying the above quantity with n . These expressions also indicate the impact of a change in the tolerance value on the number of survivors. As mentioned earlier, an increase in the tolerance values results in an increase in the number of survivors for Superstrict and DD-dominance filters. However the expression for Strict shows that there are some terms containing ϵ which increase the number of survivors while other terms decrease the number of survivors. Hence, one cannot say that the number of survivor of Strict dominance filter will necessarily result in an increase or decrease in the total number of survivors produced by application of the filter.

The pruning efficiencies of the various choice filters with respect to changing problem characteristics are shown in Figure 6 for various choice filters. Although the above expressions could have been used to compute the fraction survivors for the filters, this plot was generated by use of Monte-Carlo simulations with the trials conducted on randomly generated datasets with $n=2500$ for various values of m . This was mainly due to the computational complexity involved in evaluating the above integrals especially for large values of m and n . However, for a few points obtained by the Monte-Carlo simulations, we did compare some of the values obtained by the simulations with the values from the above expressions and they matched to a great degree of accuracy. For the simulations, the dataset was generated from a uniform distribution in the interval $[0,1]$. Each of the filters shown was then applied to the dataset with tolerances set to 0.01 and 0.1 respectively for each filter. The pruning efficiency for each filter was calculated as the mean fraction survivors over the total number of trials for each value of the tolerance.

The plots show, as expected, that Superstrict dominance produces the most number of survivors followed by ebtp-filter, followed by dd-dominance and then Strict dominance filter. However we also get to verify our additional insights as to how the pruning of each of the choice filters will be affected as the situational demands change.

Firstly, as the value of the tolerance increases, each of the three choice filters Superstrict, ebtp and dd-dominance produce more number of survivors as expected. A larger tolerance value could mean that the user is willing to compromise more on the optimality of the alternatives and thereby allow more suboptimal, or *nearly optimal*, alternatives to survive. It can be seen that for large values of tolerances, the increase in the number of survivors with respect to the number of criteria is much rapid for Superstrict than it is for ebtp and dd-dominance filter. However as m increases, the survivors produced by dd-dominance and ebtp filter tend to become identical to the Pareto set. This is because as the dimensionality of the ϵ -box increases (i.e. m increases) it becomes less and less probable for alternatives to occur in this ϵ -box. Thus neither dd-dominance nor ebtp will produce any additional alternatives in their second pass.

When the tolerance values are small, as can be seen from the plot for the 0.01

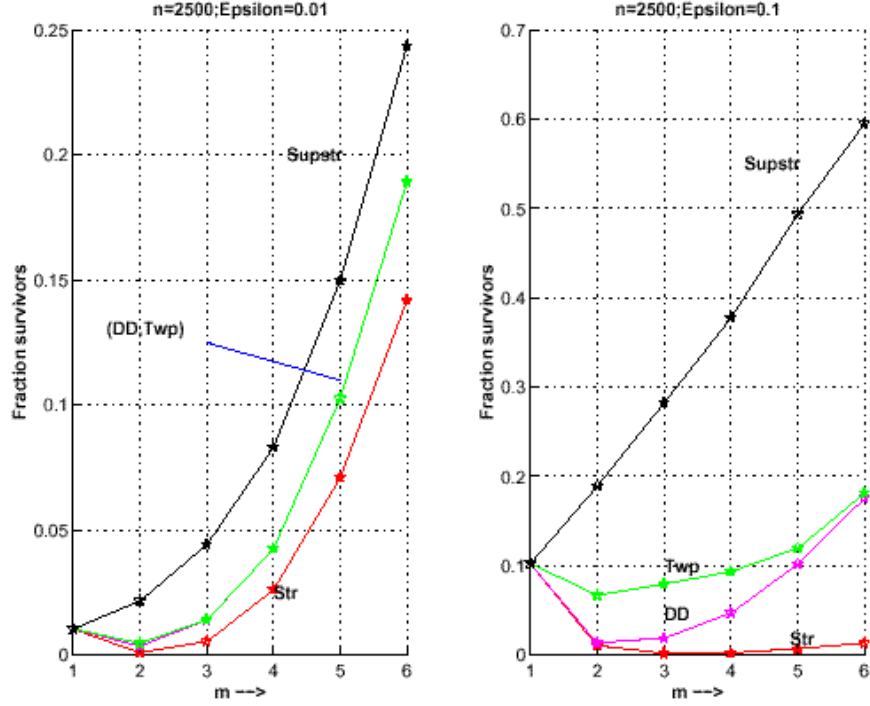


Figure 6: Plots showing fraction survivors with varying number of criteria for four different choice filters when applied to a decision problem with 2500 alternatives and with tolerances set to 0.01 and 0.1 respectively.

tolerance, both ebtp-filter and dd-dominance filter will tend to produce the same set of survivors. This is expected because of two reasons: firstly since the tolerance value is small, the resulting ϵ -box is therefore small and the chances that any alternative occurs in this small volume are considerably low. Additionally, all such alternatives which are within the small ϵ -box will consequently be closer to dominance survivors and it will be rare for them to be *dd dominated* by a dominance survivor. If no alternatives occur in some ϵ -box then both ebtp and dd-dominance produce the Pareto set as their survivors. If on the other hand, any alternatives do occur in some ϵ -box, then all such alternatives which survive ebtp will also survive dd-dominance since they will not be dd-dominated by any alternative. In either case, the survivors produced by the two

filters will tend to coincide for small values of ϵ .

3.4.2 Time complexity

An analysis of the time complexity of the dominance filter was presented in Chapter 2. We also saw the impact of the size of the dataset as well as its nature (how competitive the alternatives are, what kinds of correlations exist between the criteria) on the time for dominance filtering. In this section, we consider the time-complexities for the TBDFs discussed in the previous section.

Strict dominance filter

The Strict dominance filter uses the same algorithm as the dominance filter, as described in section 2.2.2. Therefore, similar to the analysis for the dominance filter, the average time-complexity for the strict dominance filter also depends upon the expected number of survivors for the filter. In general, this can be greater than, or fewer than, the expected number of dominance survivors depending upon the the values of the tolerances. The Monte-Carlo results show that for small tolerance values, Strict dominance filter tends to produce fewer survivors than dominance filter. Hence for small value of tolerances the running time of the strict filter is expected to be better than that of the dominance filter. However as the tolerance values increase, the expected number of survivors begin to increase thereby potentially making its running time worse than the dominance filter.

Superstrict-dominance filter

The algorithm described for the Superstrict dominance filter shows that the first pass is the dominance filter which is $O(n \cdot \ln^{m-1} n)$ as discussed in Section 2.2.2. The second pass of the filter compares each non-survivor of the first pass with the survivors of the first pass, until some survivor *superstrictly-dominates* it in which case the algorithm takes the next non-survivor from the first pass. If a non-survivor of the first pass is not *superstrictly dominated* by any of the first pass survivors, then it is marked a new survivor to be appended to the survivor of the first pass at the end of the second pass. If s be the dominance survivors produced in the first pass, maximally each of the the

$n - s$ non-survivors will require comparisons with all s survivors. This can take no more than $s \cdot (n - s)$ pairwise compares for the second pass with a time-complexity of $O(ns - s^2)$. Since the expected number of dominance survivors $s = O(ln^{m-1}n)$, the average case time-complexity of the superstrict filter, which is the sum of the time required for the first pass and the second pass is $O(n.ln^{m-1}n)$ which is the same as the average time-complexity for the dominance filter.

DD-dominance and ϵ -box-twopass filter

The algorithms for each of the ebtv filter and dd-dominance filter show that their first passes require the computation of the dominance survivors. In the second pass, each filter considers the non-survivors of the first pass and checks for the respective kind of dominance of the non-survivors by a survivor of the first pass. All such dominance non-survivors that are not *beaten* according to the respective decision rule of the filter are retained as new survivors and appended to the survivor set from the first pass. Hence, using the same analysis as that for the superstrict dominance filter from the previous section, we see that the average time complexities for both the ebtv filter and the dd-dominance filter are each $O(n.ln^{m-1}n)$.

Comparing the actual running times of the TBDFs

Since each of the Superstrict, ebtv and dd-dominance filters produce the same set in their first passes, their actual running times can be compared merely in terms of their running time during the second pass. As m increases we expect more and more non-survivors of the first pass to survive the superstrict decision rule. In fact, from Corollary 3.2 we already know that for a given value of ϵ , the superstrict filter will retain all non-survivors that are retained by the ebtv and the dd-dominance filter and potentially more. Therefore, the actual running time of the superstrict filter cannot be better than that of the ebtv, and dd-dominance filters. In fact if m or ϵ are large, the difference between the running times of superstrict filter and the other two filters above is expected to be large.

In comparing between the ebtv filter and the dd-dominance filter, firstly, if the tolerance values are small, both the filters are expected to produce the dominance

set as their survivor sets due to reasons mentioned earlier. Hence, for small values of tolerance, it is better to run the ebtp-filter since it has a greater chance of fetching additional alternatives during the second pass than the dd-dominance filter (thereby putting the second pass to good use). On the other hand if the tolerance values are large, the ebtp filter is expected to run faster than the dd-dominance filter. This is because any non-survivor that falls in the ϵ -box of a first pass survivor gets retained as a new survivor by the ebtp-filter without requiring further comparisons; however the dd-dominance will need to keep checking if this non-survivor is not *dd dominated* by any of its dominators. Of course, the ebtp filter is expected to produce a lot more survivors than the dd-dominance filter for large values of tolerances as discussed in the previous section. Finally, as the number of criteria m increases or if the tolerances become smaller, it becomes rarer for the second pass of either the dd-dominance or the ebtp filter to salvage anything at all. This is because, identical to the case when ϵ is small, the occurrence of alternatives inside ϵ -boxes becomes improbable. Hence for the same reasons as discussed above, it is better to use the ebtp filter in the interest of making good use of the second pass in fetching additional alternatives.

3.5 Selecting a choice filter as a function of problem characteristics and user needs

The above analysis of some choice filters indicates that the filtering efficiencies and the running times for the filters depend upon the characteristics of the decision problem. For example if m is large, then the difference between the number of survivors produced by the superstrict filter and that produced by the dd-dominance or the ebtp-filter is large especially since the number of survivors produced in both passes by the superstrict filter increases monotonically with increasing m , while the number of survivors produced by the dd-dominance and the ebtp-filter in the first pass is identical to that produced by the superstrict filter but the number of second pass survivors reduces monotonically with increasing m . Additionally Corollary 3.2 indicates how the filters line up in terms of their conservatism. Therefore, the choice to deploy one filter over the other is clearly a function of user-needs like expressions of conservatism, time-pressure, and problem characteristics like n , m , and ϵ . Thus the

choice of the right filter for a particular decision situation is itself an MCDM problem, with the various situational demands of the decision situation being used to decide on the best filter for the situation.

In general, the list of decision situations can be innumerable. However, even a coarsely qualitative classification of decision problems based on decision situations will reduce the problem to one of finding choice filters which are well-suited to these qualitatively defined decision situations. For the purposes of the dissertation, we classify decision problems along decision situations that are most commonly encountered. For example, at the topmost level we classify decision problems based on whether or not they have uncertainty associated with them (in criteria values). This classification is effective because the kinds of user-needs which can be addressed depend greatly upon whether or not there is uncertainty associated with the criteria values. Also, the presence of uncertainty curtails the user from expressing unachievable aspirations like produce all and only the optimal alternatives. A filter-choice schema which is aware of the presence of uncertainty can convert expressions of unachievable aspirations from the user into one of achievable trade-offs and present them to the user. The user's trade-off expressions can then be used to decide on the appropriate filter for the situation. The next chapter will make use of the performance characteristics of various filters to develop a scheme to solve the filter-choice problem for a given decision situation.

CHAPTER 4

DECISION SITUATIONS WITHOUT UNCERTAINTY

4.1 Introduction

In this chapter we will look at a class of decision problems for which the criteria values are known accurately. In the absence of uncertainty, the application of the dominance filter to a finite set of alternatives produces the Pareto set. In general, however, a decision situation might require fewer or more alternatives than in the Pareto set. In this chapter we will consider how an appropriate choice filter can be used to address the various decision situations that arise with no uncertainty present. The idea will be to make use of the performance characteristics of the different choice filters with respect to varying problem characteristics to develop a scheme for choosing an appropriate filter for a given decision situation. This scheme will enable the S-F-V-architecture to apply across a wider class of decision situations.

4.2 Tolerances to model situational demands in the absence of uncertainty

In Chapter 3, we developed a family of tolerance-based dominance filters. In this chapter, we identify situational demands that the tolerances can be used to model. Since each of the Superstrict, dd-dominance, and ebt_p filters produces the Pareto set and potentially more alternatives in addition to the Pareto set, decision situations where the DM desires more alternatives than present in the Pareto set can be met by

the use of any of these filters. In each case, the tolerances can be used to model user expressions of criteria-specific bounds on the degree of suboptimality that he is willing to tolerate. Thus, the dd-dominance filter retains the Pareto set and additionally all such suboptimal alternatives which are not worse than any of their Pareto-dominators by more than the magnitudes of the tolerances along the individual criterion. Furthermore, these bounds might be in the form of a conjunctive expression across all the criteria (namely the DM is interested in all those suboptimal alternatives which are within the tolerance bounds of their dominators along *all* the criteria), or in the form of a disjunctive expression (namely the DM is interested in those suboptimal alternatives which are within the tolerance bounds of their dominators along even a *single* criterion). The latter expression is a relatively more conservative expression where *near-optimality* along even a single criterion is considered as a desirable property of a suboptimal alternative.

The tolerances when used in Strict dominance filter can be interpreted as representing the marginal rates of substitution or trade-off rates of the decision-maker⁸. Hence when the tolerances are to be applied inside the Strict-dominance rule, the preference elicitation called for is the trade-off rates of the decision-maker. Therefore, the Strict dominance filter seems like a good filter to choose alternatives within the Pareto set. However, as argued in Chapter 1, it is often unsuitable to elicit preferential information like trade-off rates as abstract preferences; hence the use of Strict-dominance filter for further pruning is not recommended. The Viewer in the S-F-V architecture provides the user with a natural way to exert his trade-off preferences. Hence for decision situations where the DM desires fewer alternatives than in the Pareto set, we recommend the use of the Viewer.

⁸Suppose, without loss of generality, that A and B are 2-criterial alternatives such that neither dominates the other. In other words, on criterion c_1 , $(a_1 = b_1 + k_1)$ and on criterion c_2 , $(b_2 = a_2 + k_2)$. Now if we set $\epsilon_1 < k_1$ and $\epsilon_2 > k_2$ and apply *Strict-dominance* (ϵ_1, ϵ_2) , A will strictly-dominate B . One way to interpret the above ϵ 's used in Strict-dominance is to say that it modeled a trade-off expression that, *a loss of k_2 units on criterion c_2 is more than well-compensated by a gain of k_1 units on criterion c_1* so that B can be traded-off in favor of A .

4.3 The filter-choice problem for decision situations without uncertainty

The S-F-V architecture makes use of the dominance filter to produce the Pareto set with respect to the primary criteria expressed by the DM. All further pruning requirements of the user are interpreted with respect to the Pareto set as the reference set. The different choice filters we discussed in the previous chapter allow for different ways to order the suboptimal alternatives and the applicability of each filter depends upon what kinds of expressions by the decision-maker are available. Certain kinds of user-expressions call for certain basis for ordering the suboptimal alternatives and hence the application of a certain filter.

The filter-choice problem can now be seen as one of translating user-expressions of more alternatives to the exact basis upon which to order the suboptimal alternatives, and further to the choice of the filter that would produce such an ordering. This is indicated in the Figure 7. As shown in the figure, the dominance filter partitions the original set into two sets - **S** which is the Pareto set and which survives the filter and **NS** which is the suboptimal set or the set of alternatives eliminated by the dominance filter. If the DM desires fewer alternatives than in **S**, he can make use of the Viewer to further eliminate some of the Pareto alternatives in **S** by expressing various kinds of concrete preferences like trade-offs.

If, on the other hand, the decision-maker desires more number of alternatives than in **S**, use is made of an appropriate choice-filter, **CF**, from the available set of choice-filters to further partition the inefficient set, **NS**, into **NS1** and **NS2**. The set **NS1** can be seen as the survivors of the filter **CF** when applied to the set **NS**. Similarly, **NS2** can be interpreted as the non-survivors of **CF**-based filtering. Now, **NS1** is appended to the Pareto set **S** and this larger set, $(S \cup NS1)$, is presented to the decision-maker as the survivor set. If the decision-maker wishes to further explore this set to get insights or to remove a few alternatives based on some constraints, he can do this through the Viewer operating on the new set of survivors.

The success of this scheme depends on the assumption that the choice filter chosen from the library is one which, for the given problem, best satisfies the different needs expressed by the decision-maker. The manner in which this is done is the scope of the

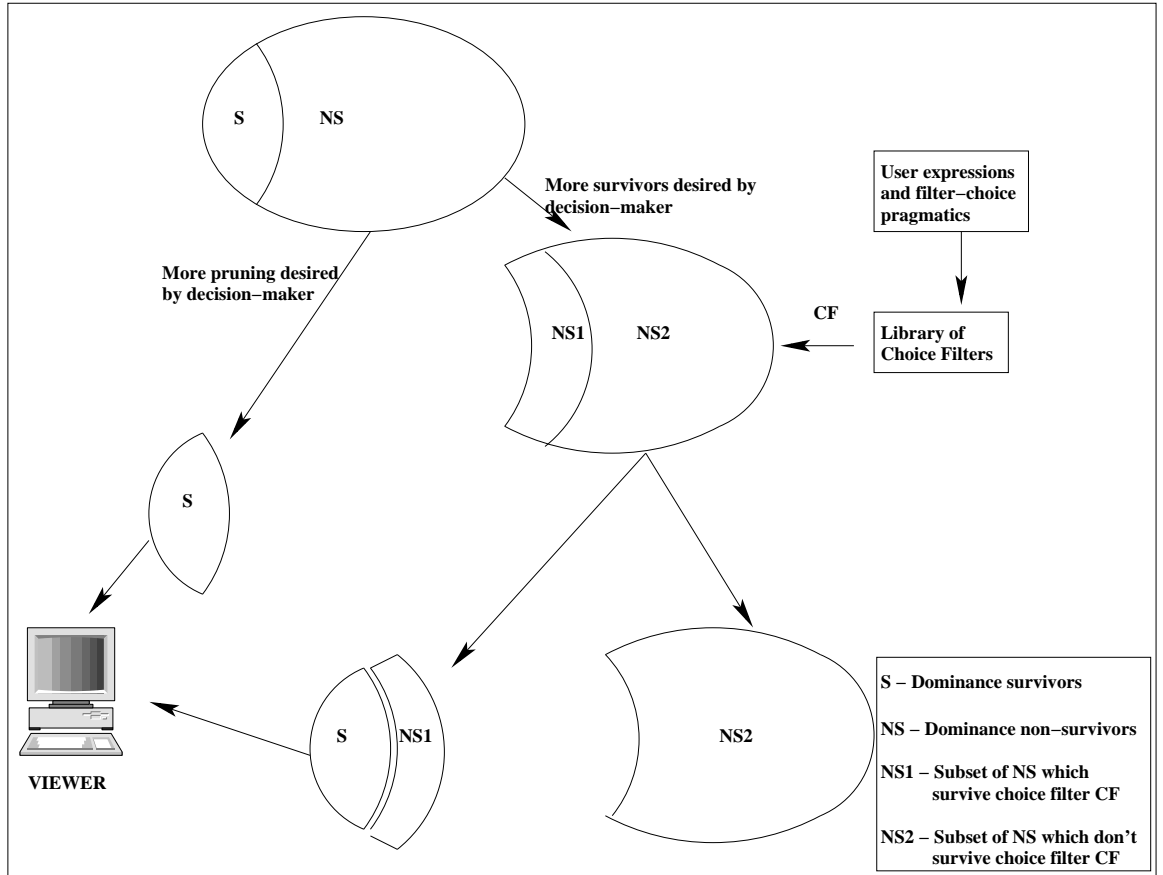


Figure 7: The filter-choice problem for situations without uncertainty.

rest of the chapter. Towards the end of the chapter, a scheme will be presented which will point to the rules or pragmatics which will be used to decide on an appropriate choice filter starting from a given set of user-needs and problem characteristics.

4.4 Pragmatics of mapping choice filters to decision situations

Based upon the performance characteristics of the various choice filters, one can now analyze the situations under which each of the above choice filters may be deployed. We only consider situations where the DM desires more alternatives than present in the Pareto set. As mentioned previously, each of superstrict, dd-dominance, and the

ebtp filter can be used to cater to the above demand. We discuss the conditions under which one of the above filters is more suitable than the other.

4.4.1 ϵ -box Twopass filter

The ϵ -box Twopass filter appends the dominance survivors with the ϵ -box survivors⁹ and reports this as its set of survivors. The question is whether these additional ϵ -box alternatives can represent some abstract pruning preference of the decision-maker. But first we examine a few conditions under which the DM could be interested in examining suboptimal alternatives:

- The DM may not be as much concerned with optimality as he is with having a suitably large number of survivors in order to get insights about the *good* regions in the structural space of the design problem (like the Genetic algorithm example mentioned earlier).
- The DM may be willing to trade-off optimality with drastic gains along the secondary criteria, or some structural characteristic of the alternative. Consider for example two alternatives, C_i and C_j , such that C_i is optimal and C_i *dominates* C_j but C_j is within the ϵ -box of C_i . Now if C_j additionally happens to perform very well in some secondary criteria, then the sub-optimality of C_j might not be as disconcerting to the user compared to the gain he obtains by virtue of C_j being much better in the secondary criterion¹⁰.

In essence, the decision-maker might be oriented towards having final solutions which might or might not be optimal in the space of primary criteria. People often choose relatively expensive commodities because of some distinguishing quality of the commodity, like color. It is unlikely that such a decision-maker does not care about optimality at all; rather he is open to choosing those sub-optimal alternatives which are very close to being optimal as long as they are

⁹Alternatives which lie inside the ϵ -box of a Pareto alternative.

¹⁰Note that this expression is not equivalent to treating the secondary criterion as a dominance criterion. The above just means that the user's sense of optimality is still related to the primary criteria he expresses. However, he is using the secondary criteria to give the *nearly-optimal* alternatives a chance to survive if they have a chance of performing exceedingly well on the secondary criterion.

outstanding in some other aspect. Such a decision-maker might express his conservatism in the following form, "do not eliminate those suboptimal alternatives which are indistinguishable from some optimal alternative within certain quantitative bounds of the criteria-vector". The decision-maker provides the bounds-vector for the m criteria and this vector can be used in the EBTP-filter to produce the newly appended survivor set.

At this point, one might wonder if the objections mentioned in Section 4.2 related to the expression of tolerances required for Strict filter would apply to the expression of tolerances for the ebtp-filter? And the answer is no, for the reasons mentioned below:

- The primary reason has to do with the exact kind of preferential information that the user is being told to give when he is asked to express the tolerances in each of the two cases. If the kind of preferential information is such that it is easy to express abstractly and moreover correctly by the user, then there is no harm done in eliciting this information in such a form. The idea of *near optimality* we contend is easier and natural to express in the form of bounds from the dominance survivors. On the other hand, people find it difficult to express trade-offs or MRSs (Marginal Rates of Substitution) abstractly and therefore the expression of tolerances for Strict filter is susceptible to the objections mentioned previously. This difference justifies the use of abstract user expression for tolerances used in ebtp-filter against the same kind of expression for use with Strict filter.
- The most common definition of the best solution to the MCDM problem is the following - *it is the solution which satisfies the decision-maker more than any of the other Pareto optimal solutions*. This deals with two independent properties, optimality which has to do with the alternatives, and satisfaction which has to do with the decision maker. It is possible for the user to be unsatisfied with an optimal solution; conversely it is possible for the user to be satisfied even if the solution is a suboptimal one. In either case, however, in order to solve for the best solution according to the above definition the user has to be allowed to minimally examine all the optimal alternatives even amidst

suboptimal ones. The use of a choice filter, like single pass Strict, to eliminate some of the optimal alternatives can result in the user not being allowed to examine them and thereby not ensuring that the best solution was solved for. On the other hand, for choice filters which minimally produce the Pareto set, even if the user is a bit conservative in his expression of the tolerance bounds on his notions of nearly optimal, we ensure that he still gets to examine all of the optimal alternatives.

These two reasons provide plausibility as to why the expression of tolerances for ebtp-filter is not objectionable on the same grounds as it is with the Strict filter.

4.4.2 Superstrict dominance filter

Similar to the ebtp-filter, the Superstrict filter retains all the Pareto survivors and potentially more. We already discussed the kinds of user-expressions for which the ebtp-filter produces a suitable survivor set by describing the properties of the suboptimal alternatives that are retained by the ebtp-filter and showing how these properties map to a class of decision-makers interested in *nearly optimal* alternatives.

The ebtp-filter brings in all such alternatives which are ϵ -close to the dominance survivors along *all* dimensions (i.e. those in the ϵ -box). As a result, an alternative that is within ϵ along $m - 1$ criteria out of the m criteria will be eliminated by the ebtp-filter. It is not hard to imagine DMs interested in such suboptimal alternatives - those which are within ϵ in at least one dimension with respect to a Pareto survivor. The Superstrict filter, in addition to all the survivors of the ebtp-filter, retains all such alternatives. Thus, if the user interest in *near optimality* is disjunctive over the criteria, then it is recommended that the Superstrict dominance filter be used for such decision situations.

4.4.3 DD-dominance filter

In Chapter 3, it was shown that both the ebtp-filter and the dd-dominance filter produce the Pareto set and additionally suboptimal alternatives which lie inside ϵ -boxes with respect to the dominance survivors. The difference between the survivors produced by the two filters is the following: the ebtp-filter produces all such suboptimal

alternatives which are in the ϵ -box of any single dominance survivor. On the other hand, the dd-dominance filter only retains a suboptimal alternative if it lies in the ϵ -box of *all* of its dominators.

If the tolerance values are small, or if m is large, it is better to use the ebt filter instead of the dd-dominance filter as mentioned before. On the other hand, if the tolerances are large, then the choice of one filter or the other depends upon whether the user-needs are oriented towards a quick decision or getting more alternatives. This is because if the tolerances are large, the ebt filter is expected to produce more survivors but do so in a shorter time since the ebt second-pass condition is easily satisfied for large tolerances. On the other hand, dd-dominance is not expected to produce as many survivors but take more time during the second pass since the dd-dominance condition will be more difficult to satisfy relative to the ebt condition for large tolerances.

4.4.4 Onion-skin filter

In general, when the user needs more than just the Pareto survivors, he might not be capable of providing the tolerances for any of the criteria. In such a case, the user might be interested in *nearly optimal* alternatives but might not be able to express this idea quantitatively using tolerances. For such decision situations, the Onion Skin filter can be used.

As mentioned in the previous section, this filter looks at the n alternatives on the m -dimensional space as groups of concentric *Pareto layers*, the outermost layer being the survivors of dominance, the next layer obtained by applying dominance to the non-survivors of dominance from the previous iteration and so on. The user's notion of *nearly optimal* is related to the number of layers that need to be peeled and the additional alternatives obtained are appended to the existing set of survivors. One way to obtain the number of peels to be applied is to get an expression from the user about the number of survivors he expects and then use that number to get an idea as to how many layers will need to be peeled in order to get that many survivors. This can be computed by using the closed form expression for the expected number of dominance survivors, given n and m . Additionally if n is large then the number of

newly added survivors from each peel will be the same as the number obtained from the previous peel in which case the user can even be allowed to express his pruning requirement in terms of multiples of the number of survivors of measured dominance (twice, thrice or k times). In such a case, k peels can be expected to approximately produce the number of survivors desired by the DM.

4.5 A scheme for the filter-choice problem in the absence of uncertainty

Summarizing on the above analysis, and based on the choice filters considered above, we are now in a position to lay out a scheme which would allow the selection of an appropriate choice filter according to the particulars of the decision situation. This scheme is shown in Figure 8 in the next page. As suggested previously, any time the decision-maker intends to get fewer survivors than produced by dominance, it is recommended that the Viewer be used for this purpose. Conversely, if the decision-maker desires more survivors than that produced by dominance, the scheme indicates how an appropriate choice filter is selected according to the situational demands. The decision-maker not only provides information about the tolerances on various criteria to express his notions of *nearly optimal* but additionally provides information about his *pruning versus conservatism trade-offs*. The scheme takes into account whether or not the decision-maker wishes to express any tolerances at all. Hence in the extreme case where the decision-maker desires more alternatives but is not in a position to express tolerances for any criteria at all, the Onion-Skin filter is used to peel the next Pareto layer which is appended to the dominance survivors.

On the other hand, if the decision-maker does express his need for more alternatives in the form of tolerances, then the choice filter selected further depends upon the existing problem characteristics and other kinds of user-needs. Thus, if the decision-maker's notion of *nearly optimal* is extremely conservative, his tolerances expressions are expected to be disjunctive over the criteria and Superstrict dominance is considered an appropriate filter for the occasion. If on the other hand, the decision-maker's idea of *near optimality* is expressed as a conjunctive expression of tolerance bounds over the criteria, then either the ebtp-filter or dd-dominance filter is used depending

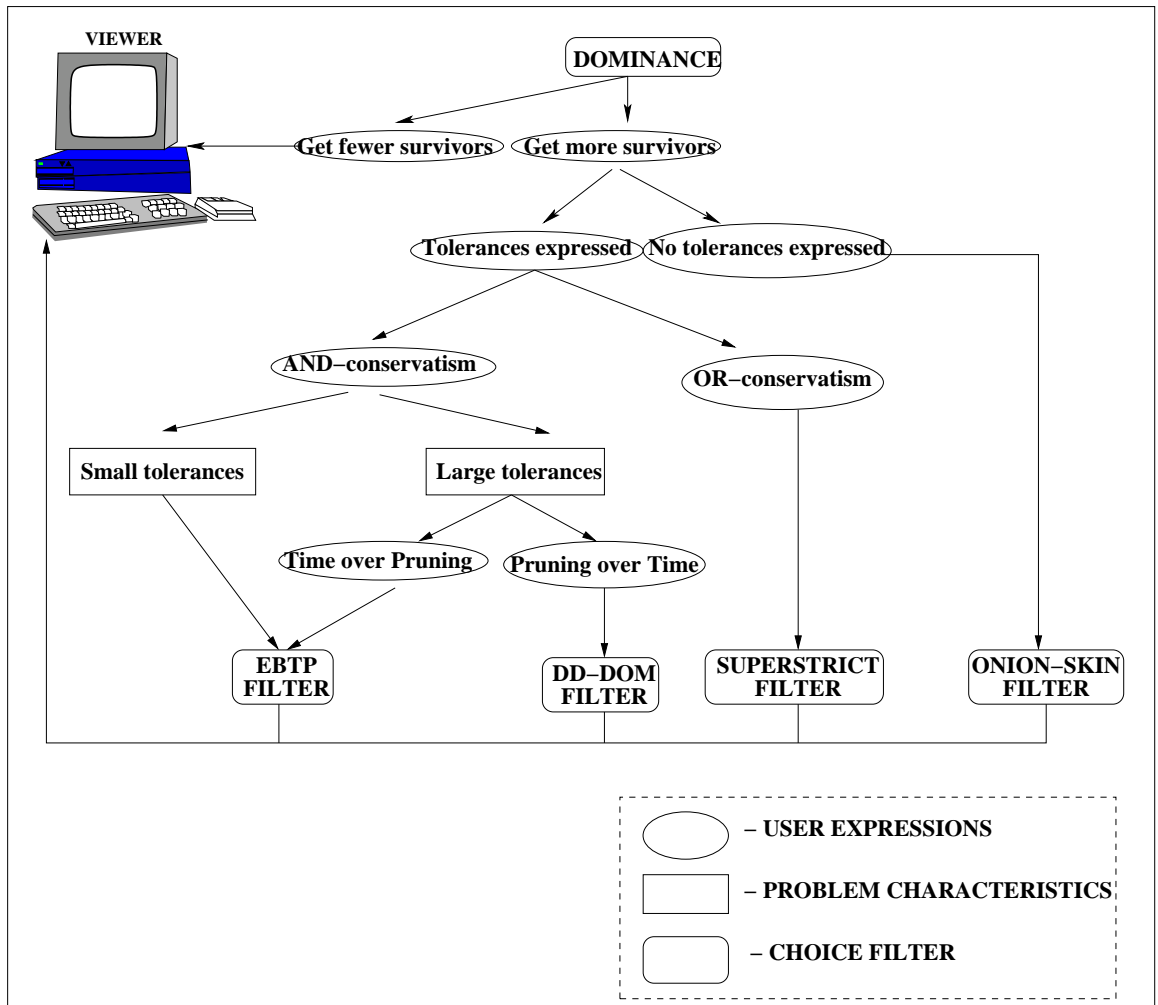


Figure 8: The filter-choice scheme for situations without uncertainty.

upon further constraints imposed by the decision-maker and the problem characteristics. If the tolerance values are small (according to the range of criteria values), then the ebtp filter is used. On the other hand, if the tolerance values are large, then if the DM desires a quicker decision but poorer pruning the ebtp-filter is used; if the DM's need for a good pruning overweighs his need for a quick decision, then dd-dominance is preferred over ebtp-filter.

Thus depending upon the existing decision situation, characterized by the problem characteristics and user-needs, an appropriate choice filter is used inside the S-F-V-architecture thereby ensuring that a better set of survivors is produced as a solution of the decision problem. In the next chapter, decision situations which have uncertainty associated with them are considered and a similar scheme showing how the filter-choice problem is solved depending upon the situational demands is developed at the end of the chapter.

CHAPTER 5

DECISION SITUATIONS WITH UNCERTAINTY

PART I: BAYESIAN CHOICE FILTERS

5.1 Introduction

In the last chapter we considered a class of MCDM problems which do not have any kind of uncertainty associated with the criteria values. In reality, most decision problems are rife with uncertainty of one kind or the other. In design optimization problems, criteria values are produced based on simulations run through computerized models; hence the accuracy of these criteria values depend upon the extent to which the models are realistic. In financial situations, the value of an asset depends upon the future outcomes of a lot of events; hence the decision to hold onto an asset or give it up is uncertain to the extent that the future cannot be predicted. In medical domains, the assignment of ailment status starting from patterns of symptoms can be inconclusive because of the inability to confirm the existence of a symptom with complete certainty. Most uncertain situations come with at least some amount of information not just about the variables of the decision problem but also about the nature of uncertainty associated with them.

In this chapter, we will discuss choice filters for dealing with uncertainty, which are modeled on Bayes' decision theory [17]. In the next 2 chapters, we will look at the use of the dominance filter, and tolerance-based dominance filters for handling decision problems with uncertainty. This will also present us with the opportunity

to compare the two classes of choice filters both in terms of their applicability to different situations and the problems associated with them.

5.2 The MCDM-problem and uncertainty

One class of MCDM-problems which result due to the presence of uncertainty of a certain kind can be formulated as below:

$$\text{maximize } \{d_{i1}, d_{i2}, \dots, d_{im}\} \text{ for } d_i \in D$$

where D is the set of n choice alternatives and d_{pq} is the value taken by the alternative d_p on criterion q . However, these values are not available; instead the *measured* criteria values, \tilde{d}_{pq} are available. These measured values are drawn from the known density functions, $f_i()$ where

$f_i()$ = the probability density function representing the uncertainty of values along criterion i ; this will commonly be referred to as the *noise model*.

The difference in this formulation compared with the one defined earlier is the presence of probability distributions for the various criteria representing the uncertainty associated with the criteria values. Also, in the above formulation the measured, but inaccurate, criteria values are available while the actual criteria values are not.

The stochastic nature of the measured criteria values allow us only to *estimate* the actual values. Therefore, any filter will be able to produce only an approximation of the actually optimal set. All filters will be prone to produce two kinds of misclassifications:

- Labeling suboptimal alternatives as optimal alternatives (*false positives*)
- Labeling optimal alternatives as suboptimal alternatives (*false negatives*)

The selection of an appropriate filter will be therefore constrained by the decision-maker's concerns with each of these two kinds of misclassifications. As a matter of notation, we will use the term $f_+(F, \tilde{\mathcal{X}})$ to refer to the false positives produced by the filter $F()$ when applied to the measured set of criteria values in $\tilde{\mathcal{X}}$. The term $f_-(F, \tilde{\mathcal{X}})$ will be used to refer to the false negatives produced by the filter $F()$ when applied to the measured set of criteria values in $\tilde{\mathcal{X}}$

Decision situations for which the decision-maker's tolerance for suboptimal alternatives is very low will require the use of a filter that produces fewer false positives. Conversely, in situations where the decision-maker might not be as concerned with having suboptimal alternatives as he is with the elimination of optimal alternatives, choice filters which produce fewer false negatives are desirable.

Typically, the use of a decision rule to reduce the number of false positives by eliminating some of its survivors will tend to increase the number of false negatives since some of the eliminated alternatives might be truly optimal but the uncertainty provides us with no information to test this. Conversely, any attempt to reduce the number of false negatives by fetching additional alternatives will result in an increase in the number of false positives. This gives an intuitive reason as to why the issue is one of trading-off since the two goals (that of producing lower false positives and lower false negatives) are conflicting.

5.3 Use of Bayesian Decision Rule(BDR) under Uncertainty

In many cases, the noise model is expressed as a density function which represents the measured values as probability distributions. For example, the noise model might be described by a Gaussian or Normal distribution with the standard deviation or the variance specified as shown in Figure 9 below. The measured criterion value, namely x_m , is a normal distribution with *mean* equal to the actual criterion value, namely x_a , and with a standard deviation σ . i.e.,

$$x_m = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{\frac{-(x-x_a)^2}{2\sigma^2}} \quad (5.1)$$

Obviously, the actual criterion value x_a , namely the *mean* of the distribution, is unavailable and the noise model is described by means of σ alone. The question is whether there is a way to use this partial information about the noise model and try and get a good approximation of the optimal set.

Provided some knowledge about the prior distributions of the actual criteria values is available, Bayes decision rule can be used to compute the posterior probabilities that each alternative is optimal. Ordering the alternatives based on the values of

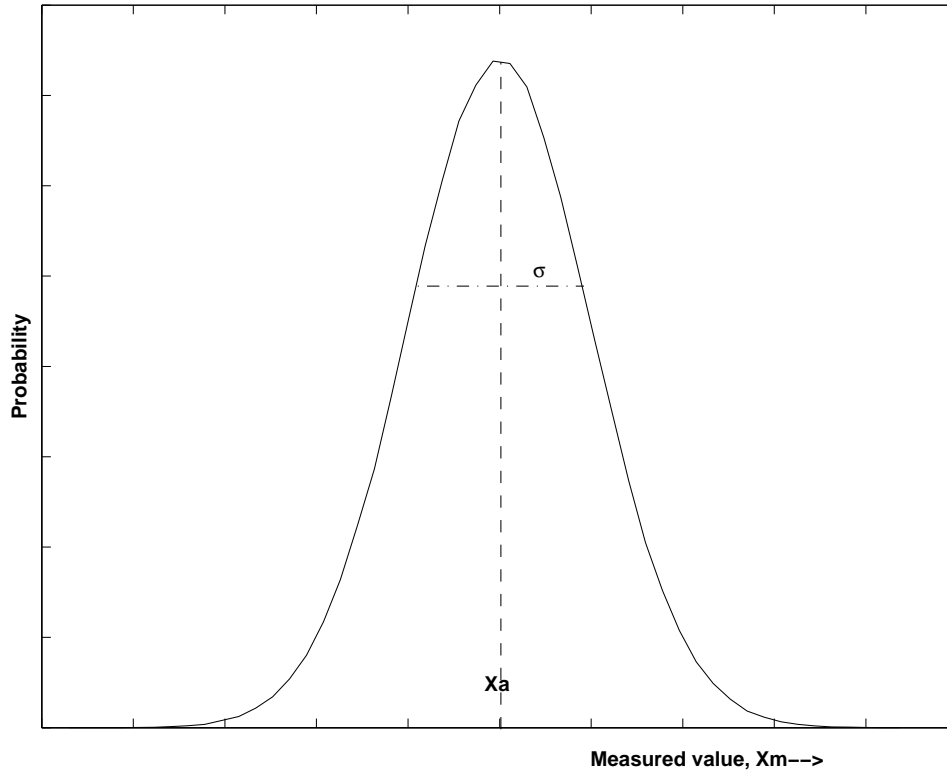


Figure 9: The Gaussian Noise Model

these probabilities will produce an ordering wherein the position of an alternative in the ordering is proportional to its chances of being optimal; we call this ordering the *Bayesian ordering*. Now, various kinds of user expressions, like the maximum amount of loss that he is ready to incur in terms of misclassifications, or the desired pruning, can all be used to select an appropriate, best subset from the ordered set. The next section shows how application of Bayesian Decision theory will allow a DSS to cater to the above kinds of user-needs in the presence of uncertainty.

Mapping user concerns through the Bayesian ordering

Let α_1 and α_2 be the possible actions of some decision rule with respect to some alternative X , where α_1 pertains to the decision to keep X while α_2 pertains to the

decision to eliminate X ¹¹. Let ω_1 and ω_2 be the true state with respect to alternative X , where ω_1 refers to the state of affairs that X is a truly optimal alternative while ω_2 pertains to the state of affairs that X is not optimal. Then, we can construct a penalty matrix as below:

$$\begin{array}{cc} & \omega_1 & \omega_2 \\ \alpha_1 & \lambda_{11} & \lambda_{12} \\ \alpha_2 & \lambda_{21} & \lambda_{22} \end{array}$$

where λ_{ij} is the penalty incurred for performing action α_i when the state of affairs ω_j is true about the alternative.

Let $R(\alpha_i|C_i)$ be the risk associated with performing action α_i with regard to alternative C_i . Then,

$$R(\alpha_1|C_i) = \lambda_{11}P(E_i) + \lambda_{12}(1 - P(E_i)). \quad (5.2)$$

$$R(\alpha_2|C_i) = \lambda_{21}P(E_i) + \lambda_{22}(1 - P(E_i)). \quad (5.3)$$

where the term $P(E_i)$ is the posterior probability that the alternative C_i is truly optimal.

If the user's interest is in obtaining just the Pareto set, we can set $\lambda_{11} = \lambda_{22} = 0$. Bayesian decision theory dictates that the optimum action to perform is the one which minimizes the posterior risk associated with the action. In other words, the decision to *keep* alternative C_i , or perform action α_1 , is an optimal¹², decision if and only if

$$R(\alpha_1|C_i) < R(\alpha_2|C_i). \quad (5.4)$$

That is, retain C_i as a survivor if and only if,

$$\lambda_{12}P(E_i) < \lambda_{21}(1 - P(E_i)). \quad (5.5)$$

¹¹i.e. the decision not to retain X as a survivor.

¹²i.e. The decision rule should keep the alternative if the total risk associated with keeping it is lesser than the total risk associated with eliminating it.

which upon simplifying yields, keep C_i as a survivor if and only if,

$$P(E_i) > \frac{\lambda_{12}}{(\lambda_{12} + \lambda_{21})}. \quad (5.6)$$

Equation 5.6 indicates that if we could order the alternatives in terms of their individual probabilities of being optimal, then expressions of cost or penalties of misclassifications as expressed by the decision maker can be used to produce a corresponding survivor set. We construct a threshold θ as follows:

$$\theta = \frac{\lambda_{12}}{(\lambda_{12} + \lambda_{21})}. \quad (5.7)$$

We now retain all those alternatives, C_r , from the Bayesian ordering for which the values of $P(E_r)$ is greater than the value of θ as obtained above. Since λ_{12} represents the penalty associated with keeping a suboptimal alternative, we can identify this term with the cost of a false positive say λ_{fp} . Similarly we can replace λ_{21} by λ_{fn} to indicate that it is the penalty associated with the event of producing a *false negative*. We can therefore rewrite the previous equation as,

$$\theta = \frac{\lambda_{fp}}{(\lambda_{fp} + \lambda_{fn})}. \quad (5.8)$$

or,

$$\theta = \frac{1}{(1 + \frac{\lambda_{fn}}{\lambda_{fp}})}. \quad (5.9)$$

If we write,

$$\mu = \frac{\lambda_{fn}}{\lambda_{fp}}. \quad (5.10)$$

we can rewrite Equation 5.9 as,

$$\theta = \frac{1}{1 + \mu}. \quad (5.11)$$

Here, μ is a user concern expressing the ratio of the penalties associated with a single false negative to a single false positive produced by a particular decision rule. We

can call μ the *conservatism coefficient* since it expresses the degree to which the user is conservative with respect to trading off false positives with false-negatives. If the value of μ is large, then the value of θ will be small and a lot of alternatives will survive from the Bayesian ordering since surpassing the threshold value is easier. Conversely, if the value of μ is small, then relatively fewer alternatives will pass the threshold and end up as survivors.

Thus, if we could obtain the Bayesian ordering, we could address different kinds of user-needs in the presence of uncertainty. Therefore it can be a very useful choice filter in the presence of uncertainty. However, it is computationally very complex to produce the Bayesian ordering. We discuss this in the next section.

5.3.1 Producing the Bayesian Ordering

Let $\tilde{\mathcal{X}}$ be the set of alternatives in terms of their measured criteria values. Suppose that the prior distributions and the noise models for all the criteria are known. In order to produce the Bayesian ordering, we need to compute for each alternative, the probability that it is optimal, given the above information. In other words, we are interested in the following event related to each alternative C_i ,

E_i = the event that C_i is optimal, given the set of measured values, the prior distributions and the noise models for all criteria.

Suppose the term $(C_i \text{ } S \text{ } C_j)$ stands for the event that alternative C_i *survives* C_j . Then, we can rewrite the event E_i as follows,

$$E_i = \bigcap_{j=1, j \neq i}^n (C_i \text{ } S \text{ } C_j)$$

The probability that the alternative C_i is optimal is the probability of the event E_i defined as above. We cannot however express the probability of the RHS as the product of the probabilities of the individual events because the individual events are not independent. This is because knowledge about the probability of the single event $C_i \text{ } S \text{ } C_j$ changes our knowledge about the distribution of the values taken by C_i . If, however, we fix the alternative C_i as taking a given set of measured criteria values,

say $M = [\tilde{x}_{i1}, \tilde{x}_{i2}, \dots, \tilde{x}_{im}]$, then the events are independent. In other words,

$$P(E_i | \tilde{X}_i = M) = \prod_{j=1, j \neq i}^{j=n} P(C_i \text{ } S \text{ } C_j | \tilde{X}_i = M) \quad (5.12)$$

where each individual probability can be expressed as,

$$\begin{aligned} P(C_i \text{ } S \text{ } C_j | \tilde{X}_i = M) &= 1 - P(C_j \text{ dominates } C_i | \tilde{X}_i = M). \\ &= 1 - P\left(\prod_{k=1}^{k=m} (x_{jk} > x_{ik}) | \tilde{X}_i = M\right). \end{aligned}$$

If we assume the criteria to be independent we can write,

$$P(C_i \text{ } S \text{ } C_j | \tilde{X}_i = M) = 1 - \prod_{k=1}^{k=m} P(x_{jk} > x_{ik} | \tilde{X}_i = M). \quad (5.13)$$

From Equations 5.12 and 5.13 we can therefore, write,

$$P(E_i | \tilde{X}_i = M) = \prod_{j=1, j \neq i}^{j=n} \left[1 - \prod_{k=1}^{k=m} P((x_{jk} > x_{ik}) | \tilde{X}_i = M) \right]. \quad (5.14)$$

Since, we know the prior distributions and the noise distributions, we can use Bayes' rule to compute the probability densities $f_q(x_{pq} | \tilde{x}_{pq})$ along each of the m criteria. We are now ready to express the probability of the event E_i that we are interested in. In other words, we can now make use of Equation 5.14 to write,

$$\begin{aligned} P(E_i) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \overbrace{\dots}^m \int_{-\infty}^{\infty} \prod_{j=1, j \neq i}^{j=n} \left[1 - \prod_{k=1}^{k=m} P(x_{jk} > x_{ik}) \right] \cdot f_1(x_{i1} | \tilde{x}_{i1}) f_2(x_{i2} | \tilde{x}_{i2}) \\ &\quad f_3(x_{i3} | \tilde{x}_{i3}) \cdots f_m(x_{im} | \tilde{x}_{im}) dx_{i1} dx_{i2} \cdots dx_{im}. \end{aligned} \quad (5.15)$$

Now, each probability term on the RHS is a pairwise comparison along a single criterion and can be computed as,

$$P(x_{jk} > x_{ik}) = \int_{x_{ik}}^{\infty} f_k(x_{jk} | \tilde{x}_{jk}) dx_{jk}. \quad (5.16)$$

Or, we can now write Equation 5.15 as,

$$P(E_i) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \overbrace{\dots}^m \int_{-\infty}^{\infty} \prod_{j=1, j \neq i}^{j=n} [1 - \prod_{k=1}^{k=m} \int_{x_{ik}}^{\infty} f_k(x_{jk} | \tilde{x}_{jk}) dx_{jk}] \cdot f_1(x_{i1} | \tilde{x}_{i1}) f_2(x_{i2} | \tilde{x}_{i2}) f_3(x_{i3} | \tilde{x}_{i3}) \cdots f_m(x_{im} | \tilde{x}_{im}) dx_{i1} dx_{i2} \cdots dx_{im}. \quad (5.17)$$

Computational Complexity of producing the Bayesian Ordering

Equation 5.17 expresses the probability that a single alternative C_i is optimal, given the measured values, the prior distribution and noise. This quantity is required to be computed for each of the n alternatives in order to produce the Bayesian ordering. A user-expressed value of θ can then be used to retain only those alternatives for which the above probability is greater than θ .

Equation 5.17 also indicates the complexity involved in producing the Bayesian ordering. The first product ranges over $n - 1$ alternatives and therefore the overall work involves n^2 considerations minimally in order to produce the probabilities for all the alternatives. The integrals make it additionally complex. There are m single integrals involved for each of the n^2 considerations. And finally there are m outermost integrals for each of the n alternatives. All of the above restricts an easy application of Bayesian ordering scheme for producing the survivor set of interest to the DM.

5.3.2 Summary

The computational complexity of producing the ordering is the primary bane in the application of a filter based on Bayesian scheme described in this chapter. Another problem with the Bayesian technique has to do with the parameter μ which allows the decision-maker to specify his conservatism. This expression does not allow for differentiation among different kinds of false positives and false negatives. For instance, the term λ_{fp} in Section 5.3 is supposed to equally capture the user's loss associated with *all false positives*. However, it is very likely that a decision-maker finds false positives of a certain kind more bearable than another kind. For example, false positives that are in the neighborhood of Pareto alternatives need not be as worrisome to the DM as

the ones that are farther away from the optimal alternatives. The BDR-based filter has no mechanism to accommodate these distinctions.

A final problem has to do with the classical objection against the Bayes methodology in general - regarding the availability of the kinds of knowledge required for applying Bayes' rule. For many decision problems, complete knowledge about the probabilistic structure of the problem might not be available. For example, the specification of the noise model might not be as specific as a normal distribution with all parameters specified. Weaker noise models could be specified using terminology like, *central-tendency models* with or without any parameters specified. In such a case, it is not clear how Bayesian techniques can be used. In the next chapter, we will look at how tolerance-based dominance filters can be used to tackle at least some of the problems with the BDR-based filter mentioned in this chapter for certain class of problem distributions and noise models. The desired result is that the expressiveness of the BDR-based filter be preserved by the use of TBDFs but in exchange computational tractability is achieved along with robustness in the face of insufficiently specified noise models.

CHAPTER 6

DECISION SITUATIONS WITH UNCERTAINTY

PART II: TOLERANCE BASED DOMINANCE FILTERS

6.1 Introduction

In the previous chapter, we discussed how Bayesian Decision Theory can be used for decision problems with uncertainty. At the end of the chapter, we also discussed some of the problems involved in using the Bayesian ordering technique (primarily its computational complexity). In this chapter, we will show how different TBDFs can be used to tackle decision problems with uncertainty. In order to do this, we find it useful to classify problems based the following two kinds of uncertainty models:

1. *bounded uncertainty* where the distribution of the actual value is such that it is bounded on both sides of the measured value, i.e, $\tilde{x}_{ij} - \delta \leq x_{ij} \leq \tilde{x}_{ij} + \delta$, and
2. *unbounded uncertainty* where the distribution of the actual values around the measured values is not bounded and is of the form $x_{ij} = f(\tilde{x}_{ij})$, where \tilde{x}_{ij} is the expected value of x_{ij} and $f()$ is the unbounded probability density function.

If the interest is in producing the optimal set, we want a choice filter to produce no false positives or false negatives. However, in the presence of uncertainty, such a filter may not be possible. In this case, the best filter might be one that respects the DM's tradeoffs between false positives and false negatives. In other words, the

FP-FN characteristics of a choice filter (plot of the expected number of false positives versus the expected number of false negatives produced by the choice filter) can be used to assess its performance. In this chapter, we present the FP-FN performance of the TBDFs. We often refer to the FP-FN curve to refer to the above characteristic. But first, we discuss the performance of measured dominance for each of these two classes of decision problems.

6.2 Performance of measured dominance for bounded noise models

One way to model uncertainties in the criteria values is to provide bounds, in the form of tolerances, on the criteria values. These tolerances could be tight bounds, restricting the actual values to within a range of the measured values. For example, for a given measured value say \tilde{x} , the model is said to produce an evaluation that is accurate or reliable within $\pm\delta$ from the measured value. In other words, the actual evaluation can be expected to lie in the range $[\tilde{x} + \delta, \tilde{x} - \delta]$. This is a specific, restricted type of uncertainty model for which stronger characterizations can be given about the behaviors of the TBDFs. For decision problems with the above kind of noise model, in treating the measured value of an alternative on a particular criterion as its actual value, the magnitude of error associated with the judgment can be easily assessed by using the known bounds. For example it can be inferred that the value \tilde{x} could maximally have been as large as $\tilde{x} + \delta$ and no smaller than $\tilde{x} - \delta$.

Suppose two measured values are compared and we infer $x > y$ based on $\tilde{x} > \tilde{y}$, then if it happened to be the case that $y > x$, we must still have $y \leq x + 2\delta$ so that the error in declaring $x > y$ is less than 2δ . Hence, when measured dominance uses such pairwise comparisons across all the criteria, the error associated in declaring that an alternative C_i dominates another alternative C_j is similarly bounded in the worst case by 2δ along all criteria. In other words, suppose in the above case, C_j actually dominated C_i , we could still assert that C_i would still be within 2δ on all criteria with respect to its dominator C_j . More generally, for all survivors of measured dominance one could say that the suboptimal alternatives are no worse than 2δ smaller in criteria evaluations with respect to their actual dominators. If this difference of

2δ is not critically significant to the decision-maker, then the survivor set can be considered as a best solution to the problem. Similarly, it can be said that for any optimal alternative eliminated by the measured dominance filter for the bounded noise model, it could not have been better by more than 2δ in terms of its actual evaluations with respect to its measured dominators. Therefore, for many practical problems, if the noise model is bounded, the application of measured dominance and many of its advantages, including its filtering efficiency still apply.

6.3 Performance of measured dominance for unbounded noise models

In this case, the actual values are described by probability density functions which are unbounded around the measured criteria values. Consider a random variable X , with density function $f(x)$ and distribution function $F_X(x)$. Let X represent the actual value taken by an alternative on some criterion. Now suppose the actual value taken by another alternative on the same criterion is of the form $Y = X + \delta$. Then it is well-known that the random variable Y has a density function $g(y) = f(x - \delta)$ and distribution function given by $G_Y(y) = F_X(x - \delta)$. In other words, the probability functions for two actual values along a single criterion can be obtained by a *shifting* operation, the amount and direction of shifting being dependent upon the magnitude and sign of δ respectively. The family of distributions so obtained is defined in probability theory as the *location family* of distributions with respect to the given distribution and δ is commonly referred to as the location parameter. The following are well-known results for the location family of distributions: For X and Y described as above,

$$E[Y] = E[X] + \delta. \quad (6.1)$$

$$G^{-1}(p) = F^{-1}(p) + \delta, \text{ for } p \text{ in } (0, 1). \quad (6.2)$$

In our case, since the expected values of the variables are the measured criteria values, we can also rewrite Property 6.1 as $\tilde{Y} = \tilde{X} + \delta$. Property 6.2 indicates that the cumulative distribution functions of the two random variables are also separated by δ units. Location families can be associated with many well-known distributions; the

normal distribution, the uniform distribution, the exponential distribution are some examples.

Since the noise model is assumed to be unbounded, for a given measured value the corresponding actual value could lie anywhere in the range $[\infty, -\infty]$. However, we will show that different kinds of assurances can be provided to the DM by the use of appropriate TBDFs even if the noise model is unbounded. In this section, we present some theorems related to the survivors of measured dominance for decision problems with unbounded noise models. We begin with the following theorem which is true for the measured dominance survivors.

Theorem 6.1 *If the actual criteria values and the measured criteria values belong to continuous distributions then the application of measured dominance produces a survivor set for which the following is true:*

$$E(|f_+(D, \tilde{\mathcal{X}})|) = E(|f_-(D, \tilde{\mathcal{X}})|) \quad (6.3)$$

Proof: See Appendix A.

This theorem is useful for scenarios where the DM is unable to provide his tradeoffs along the FP-FN curve. In such a case, one good solution would be to provide the DM with a survivor set by assuming that the DM's penalty associated with producing a false positive is the same as that associated with producing a false negative. Theorem 6.1 shows that the measured dominance set is such a survivor set.

It will be shown later that measured dominance can be seen as belonging to, or as a special instance of, a family of tolerance-based dominance filters. The measured dominance set is special in the sense that it operates on the assumption that the two penalties are equal. For problems where the DM expresses different penalties for a false positive and a false negative, other filters in the family can be used. The measured Pareto set can be thus used as a reference point to apply trade-offs in either direction¹³ on behalf of the decision-maker. Operationally, the set of measured dominance survivors can therefore serve as a good starting set for all decision problems with uncertainty upon which further filtering operations can be applied to add or remove alternatives as desired.

¹³Fewer FPs or Fewer FNs

The following can also be proved for the survivor set produced by measured dominance.

Theorem 6.2 *For location family noise models, if alternative C_i dominates alternative C_j in measured dominance then the probability that C_i is not actually dominated by C_j is greater than the probability that C_j is not actually dominated by C_i . In other words,*

$$[C_i \text{ measured dominates } C_j] \rightarrow [P_j(E_i) > P_i(E_j)]$$

where $P_j(E_i)$ is the probability that alternative C_i is not actually dominated by the alternative C_j .

Proof: See Appendix A.

Theorem 6.3 *For location family noise models, if alternative C_i is a survivor of measured dominance then,*

$$\forall j \ P_j(E_i) > 0.5.$$

where $P_j(E_i)$ is the probability that the alternative C_i is not actually dominated by alternative C_j

Proof: See Appendix A.

Theorem 6.3 states that for all survivors of measured dominance, the probability that they will survive with respect to any of the other alternatives is greater than 0.5. Thus, for every measured dominance survivor, there is more than an even chance probability that it will survive each possible pairwise comparison. The application of measured dominance does not require any information about the noise model since it operates only on the measured set of criteria values. The above theorems show that as long as the desire is to produce the optimal set, applying measured dominance produces a set of survivors which can be useful for many kinds of user-needs with respect to optimality, even in the presence of uncertainty.

In summary, for decision situations with uncertainty and under various conditions, the dominance filter can still be used as an effective choice filter to meet the demands of the decision situation. Specifically, for decision problems with bounded noise models, precise bounds can be provided on the degree of sub-optimality of the false positives retained, and on the degree of optimality of the false negatives eliminated by

the dominance filter when applied to measured criteria values. Decision-makers for whom the associated losses based on these bounds are tolerable can therefore make a choice based on the measured dominance set. Also, if the DM is unable to express his FP-FN tradeoffs, then the measured dominance set represents a set based on assigning equal penalties to the production of a false positive and a false negative.

6.4 TBDFs for bounded noise models

In this section, we will first consider the bounded noise model and see how tolerance-based dominance filters can be used to provide different kinds of assurances and thereby cater to different kinds of user demands in the face of uncertainty. Consider a bounded noise model with the measured value, \tilde{x} along with bounds on both sides of the evaluation, restricting the range of the actual values that correspond to this evaluation. We represent such a model as $M(\delta)$ to reflect the fact that the actual value is bounded on both sides of the measured value by δ units. Without loss of generality, we represent the magnitude of the bounds on both sides by the quantity $\epsilon/2$. This only means that we use the variable ϵ to be 2δ where the bounds indicated for the model are $\pm\delta$. In other words, we have following m-vector specifying the model tolerances for each of the m criteria, $\{\pm\delta_1, \pm\delta_2 \cdots \pm\delta_m\}$. From this we obtain the decision rule tolerances, $\{\pm\epsilon_1, \pm\epsilon_2 \cdots \pm\epsilon_m\}$, where $\epsilon_i = 2\delta_i$. The reason for this representation will soon become clear. Note that whereas typically the rule-tolerances are treated as parameters of a TBDF rule, here the tolerances are fixed according to the model bounds. In other words, in this section when we refer to a TBDF, F , we implicitly refer to $F(2\delta)$.

Now suppose the above measured value, \tilde{x} is compared with another measured value, say \tilde{y} . Ideally, for each pairwise comparison we want to be able to use measured values, \tilde{x} and \tilde{y} , and infer the order relation between the actual values, x and y . For the bounded noise model, it is possible to do this for certain comparisons. We do this by using worst-case assumptions for one measured value and the best-case for the other. In other words, for the bounded noise model, we know that the smallest value that \tilde{x} can actually correspond to is $\tilde{x} - \epsilon/2$. It cannot be any lower because of the bounded model. Similarly, the largest actual value that can correspond to the

measured value \tilde{y} is $\tilde{y} + \epsilon/2$. So we can say that if the smallest possible value that \tilde{x} can actually correspond to, is greater than the largest possible value that can be actually taken by the evaluation \tilde{y} , then this implies that under all circumstances, we must have $x > y$. This implies that in order to ensure that $x > y$, we need to ensure that:

$$\tilde{x} - \epsilon/2 > \tilde{y} + \epsilon/2$$

or that,

$$\tilde{x} > \tilde{y} + \epsilon$$

In other words, for the bounded noise model,

$$\tilde{x} > \tilde{y} + \epsilon \rightarrow x > y \quad (6.4)$$

If the left hand side of Equation 6.4 is satisfied, then we can be sure that the alternative taking the measured value \tilde{x} above could not have been dominated by its counterpart taking the measured value \tilde{y} above. Each of the respective decision rules for Strict dominance, DD-dominance and Superstrict dominance require the above expression to be true along at least one criterion, as a condition of their respective kind of ϵ -dominance. Therefore for each of these TBDFs, if there are two alternatives C_1 and C_2 such that C_1 ϵ -dominates C_2 according to the corresponding TBDF-rule then, for a bounded noise model, we can infer that C_2 could not have actually dominated C_1 . This is because the first clause in each of the above TBDF-rules ensure that there is at least one criterion in which C_1 is known to be *actually better* than C_2 . Therefore, the dominance rule will not allow C_2 to dominate C_1 .

Since the noise model is probabilistic, it is still not possible for the general case to produce a single set of survivors containing *all and only*, i.e. the *exact* set of, optimal alternatives by the use of some decision rule. However, it will be shown next that one can make use of Equation 6.4 to solve the decision problem towards providing some strong assurances towards the ideal goal of producing *the exact* set of optimal alternatives. More specifically, and for the bounded noise model, Equation 6.4 along with two decision rules or TBDFs allows us to produce the following two survivor sets:

- The *sufficient set* is the set of survivors which is guaranteed to contain *all* the optimal alternatives and additionally guaranteed to maximally eliminate suboptimal alternatives. By maximal elimination we mean that all and only alternatives for which it can be categorically shown that they are suboptimal will not be included in the sufficient set. We use PS to denote the sufficient set.
- The *necessary set* is the set which can be guaranteed to eliminate *all* suboptimal alternatives and additionally guaranteed to maximally contain optimal alternatives. By maximal containment we refer to the fact that all and only alternatives for which it can be categorically shown that they are optimal will be retained by the necessary set. We denote this set by NS .

Obviously, the *necessary set* will be a subset of the *sufficient set*. The *sufficient set* corresponds to the class of users who are concerned about losing even a single optimal alternative i.e., those who desire a false negative rate of zero. At the other extreme, the *necessary set* can cater to the class of users who are worried about being exposed to choosing a suboptimal alternative i.e., those who desire a false positive rate of zero. The survivor set corresponding to the class of users in between these two extreme stances will still be a subset of the sufficient set and contain the necessary set, but it will be smaller than the sufficient set and larger than the necessary set. All of this is indicated in Figure 10. The figure shows three different survivor sets and each corresponds to three classes of users or user-concerns. The set of survivors indicated as the *in-between set* applies to a class of users for which the sufficient set might be too large and the necessary set too small. Assuming that the primary concern of this class is optimality, the task becomes one of appending the *necessary set* with some of the alternatives from the *sufficient set* which are not in the necessary set but which can be known to have a high chance of being true positives. The next two claims both specify and prove how the sufficient set and necessary set can be produced.

Theorem 6.4 *For bounded noise model $M(\epsilon/2)$, the application of Superstrict dominance filter $SS(\epsilon)$ produces the sufficient set*

Proof : See Appendix A.

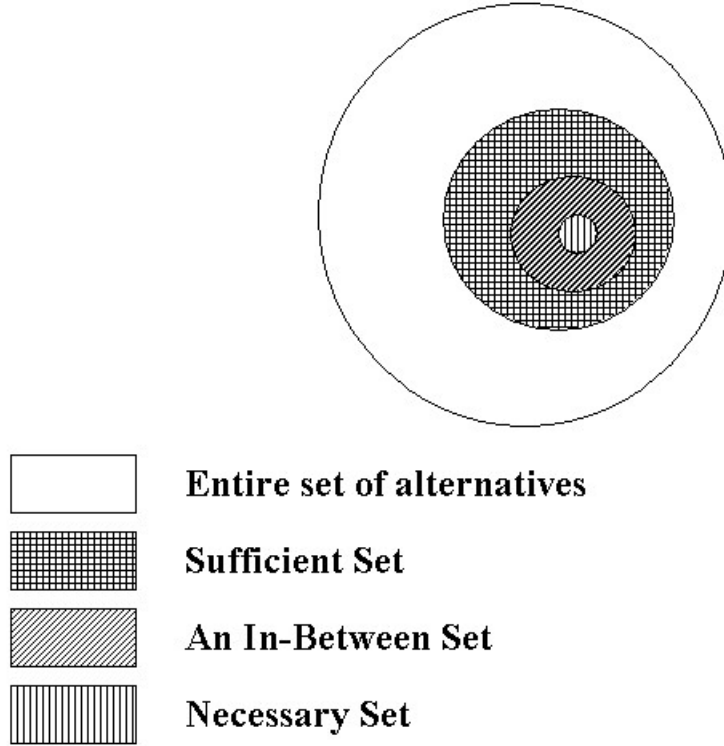


Figure 10: An illustration of the sufficient, necessary, and an in-between set of survivors corresponding to different classes of decision-makers.

The Superstrict dominance rule, in conjunction with Equation 6.4 indicates that every survivor of the Superstrict filter is such that it is better (actually) than or equal to every non-survivor along all the criteria, and strictly better than every non-survivor along at least one criterion. Since this is exactly the dominance rule in terms of the actual values, the survivors of Superstrict will not eliminate any true positive. Thus it produces the *sufficient set*. However by virtue of its rule, it will also allow for the survival of alternatives which are false positives. To see this, it is sufficient to consider a counterexample consisting of two measured Pareto alternatives C_i, C_j such that C_i is ϵ -better than C_j on some criterion but C_j is not ϵ -better than C_i on any criterion. Clearly, both these alternatives will survive the Superstrict filter since they

belong to the measured Pareto set, according to Corollary 3.3. However, since there are no criteria for which C_j is ϵ – *better* than C_i , it is quite possible C_j would have been dominated by C_i . However, due to ϵ – *indistinguishability* along some of the criteria, there is no way to infer this by using Equation 6.4. Hence, the Superstrict filter can produce some false positives as well. Theorem 6.4 also allows us to see that the measured dominance set is a subset of the sufficient set

The *necessary set* is bit more complicated to produce. Essentially, we need a survivor set such that every survivor will be ϵ – *better* than every non-survivor along some criterion or the other. In such a case, using Equation 6.4 we can infer that no non-survivor could have dominated any of the above survivors. A filter which can produce such a survivor set will have produced a *necessary set*. Theorem 6.5 shows that the Strongly-Strict filter produces the necessary set.

Theorem 6.5 *For bounded noise model $M(\epsilon/2)$, the Strongly Strict filter $Sts(\epsilon)$ produces the necessary set.*

Proof: See Appendix A.

Each survivor of the Strongly Strict filter is such that it is ϵ -better than every other alternative on some criterion or the other. For a bounded noise model, this means that each survivor is *actually* better than every other alternative on some criterion or the other, which is the same as saying that each survivor of the Strongly Strict filter is a true positive. This filter will generally not be able to produce *all* of the optimal alternatives, i.e. it will produce false negatives. This is easy to understand by considering the weakly-tied survivors of Twopass strict that are eliminated by this filter. Among themselves, these weakly-tied alternatives could either both be optimal or one could dominate the other. But since they are ϵ – *indistinguishable* along all criteria, there is no way to infer the above by using Equation 6.4. Hence, the Strongly Strict filter can produce false negatives.

The ability of the Strongly-Strict filter to produce the necessary set suggests a way to annotate the survivors of Superstrict filter with the subset which is the necessary set. We already know that the survivors of the Superstrict filter contain all measured dominance survivors, and hence all the survivors of the Strongly Strict filter (from Corollary 3.1). This is especially useful with the Viewer in the S-F-V architecture.

The Viewer can be used to not only display the survivors of Superstrict filter which contain *all* of the optimal alternatives, but additionally the subset within this set which is known to be surely optimal can annotated as using a different color. This can provide the decision-maker with a lot of information that he can use to make further choices in the Viewer.

6.5 TBDFs for unbounded noise models

In this section, we examine the applicability of TBDFs to decision problems for which the actual values are described as unbounded distributions around the measured values. Although the actual value can lie anywhere in the range $[\infty, -\infty]$, for many distributions, the probability that the actual value will be different from the measured value drops off quite rapidly in proportion to the difference between the actual value and the measured value. For example, for the normal noise model, the probability that the actual value will be outside the range $[\tilde{x} - 2\sigma, \tilde{x} + 2\sigma]$ is known to be less than 0.05. In such a case, it becomes possible to treat the normal distribution as a bounded noise model with the distribution set to zero beyond the above range on both sides. And now the decision problem can be treated as in the previous section and the necessary and sufficient sets can be produced as before.

However, since there is a non-zero probability that the actual value can indeed fall outside this approximated bounded range, the kinds of assurances on the sufficient and necessary set become weaker. In other words, the necessary set can no longer be assured to contain *only* optimal alternatives with a probability of 1.0, and the sufficient set can no longer be guaranteed to contain *all* optimal alternatives. The notions of necessity and sufficiency can still be applied as long as we weaken the guarantees associated with these sets for the bounded noise case. We introduce the terms *near-sufficiency* and *near-necessity* to refer to the corresponding sufficient and necessary sets that are produced by bounding the unbounded distributions. Since there is more than one way to bound the unbounded distributions, each way of bounding will produce its own pair of *nearly-necessary* and *nearly-sufficient* sets. These notions become especially useful if we can additionally place bounds on the probability that nearly-necessary set contains a suboptimal alternative, and on the probability that

the nearly-sufficient set does not contain an optimal alternative. Before we obtain such bounds, we first show how user expressions of conservatism can be mapped to tolerances for location family, noise models.

6.5.1 Mapping conservatism expressions to tolerances

Consider the quantity $P(x_{kp} > x_{ip} | \tilde{x}_{kp} > \tilde{x}_{ip})$. This is the probability that the actual value of alternative C_k on criterion p is greater than the actual value taken by alternative C_i on the same criterion, given that the corresponding relationship holds for the measured values. The measured values are assumed to represent the expected values. Suppose the measured values are separated by δ units along this criterion. In other words, $\tilde{x}_{kp} = \tilde{x}_{ip} + \delta$, i.e. δ is the location parameter.

Let $F(x)$ be the cumulative distribution function for x_{ip} and $G(x)$ be the cumulative distribution function for the variable x_{kp} . Figure 11 shows the two distribution functions; they are separated by δ units, with $G(x)$ being shifted δ units to the right of $F(x)$ since $\delta > 0$. Now, we can write

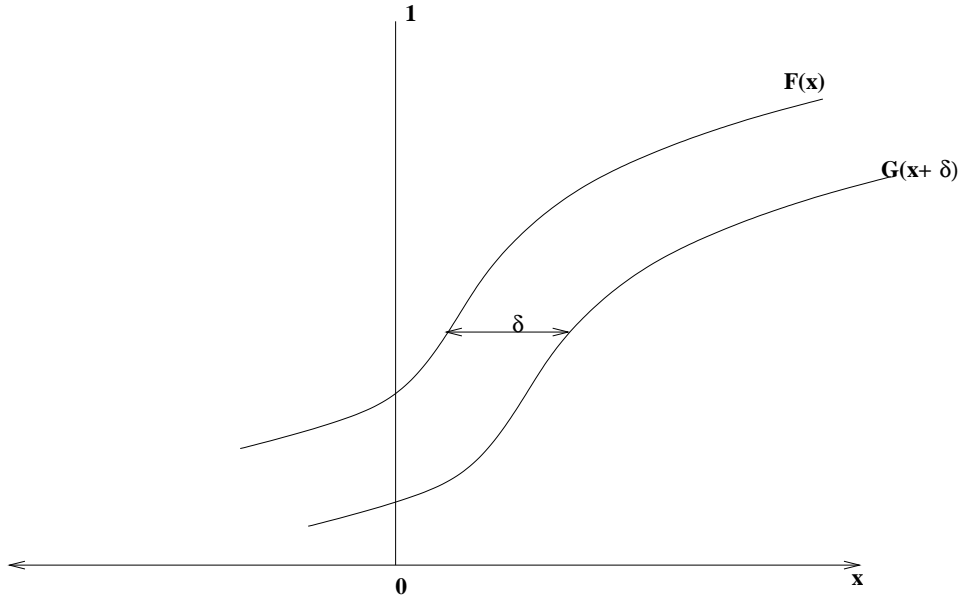


Figure 11: Comparing two distributions in the location family.

$$P(x_{kp} > x_{ip}) = 1.0 - \int_{-\infty}^{\infty} G(x_{ip} - \delta) f(x_{ip}) dx_{ip} \quad (6.5)$$

Also, for $\delta > 0$,

$$G(x_{ip} - \delta) < F(x_{ip}).$$

Therefore,

$$\int_{-\infty}^{\infty} G(x_{ip} - \delta) f(x_{ip}) dx_{ip} < \int_{-\infty}^{\infty} F(x_{ip}) f(x_{ip}) dx_{ip}. \quad (6.6)$$

Now the right hand side of Equation 6.6 above is 0.5. Therefore, we can rewrite the equation as,

$$\int_{-\infty}^{\infty} G(x_{ip} - \delta) f(x_{ip}) dx_{ip} < 0.5. \quad (6.7)$$

Equations 6.5 and 6.7 together imply that,

$$P(x_{kp} > x_{ip}) > 0.5. \quad (6.8)$$

Conversely, if $\delta < 0$ (i.e., $\tilde{x}_{ip} > \tilde{x}_{kp}$) then we get,

$$P(x_{kp} > x_{ip}) < 0.5. \quad (6.9)$$

In other words, we can write

$$P(x_{kp} > x_{ip}) = M(\delta, p) \quad (6.10)$$

where $M()$ is a monotonically nondecreasing function of δ , the location parameter and p refers to the other parameters of the distribution. Also,

$$\begin{aligned} M(\delta, p) &> 0.5 \text{ for } \delta > 0. \\ &< 0.5 \text{ for } \delta < 0. \\ &= 0.5 \text{ for } \delta = 0. \end{aligned} \quad (6.11)$$

This implies that a comparison between two actual criteria values can be expressed in the form of a monotonically nondecreasing function of the difference between the available measured criteria values, if the $M()$ function can be obtained from the

description of the noise model distribution. Using this function, probabilistic assurances in terms of the actual criterion values can be translated in terms of the available measured values.

To illustrate this, suppose the noise model is a normal distribution. Then, we have,

$$P(x_{kp} > x_{ip} | \tilde{x}_{kp}, \tilde{x}_{ip}) = P(x_{kp} - x_{ip} > 0 | \tilde{x}_{kp}, \tilde{x}_{ip}). \quad (6.12)$$

We know the difference of two random variables with normal distributions, $N(\mu_1, \sigma)$ and $N(\mu_2, \sigma)$ is also normal with mean, μ_d , and standard deviation, σ_d , expressed as,

$$\mu_d = \mu_1 - \mu_2$$

$$\sigma_d = \sigma\sqrt{2}$$

Coming back to Equation 6.12, since both the actual values are normally distributed around their respective measured values (noise models are normal), the probability can be rewritten as

$$P(x_{kp} - x_{ip} > 0) = P\left(\frac{x_{kp} - x_{ip} - (\tilde{x}_{kp} - \tilde{x}_{ip})}{\sigma\sqrt{2}} > \frac{\tilde{x}_{ip} - \tilde{x}_{kp}}{\sigma\sqrt{2}}\right)$$

This standardizes the resultant Gaussian¹⁴. The above quantity can therefore be computed in terms of the distribution function for the standard normal, $NCDF$. In other words,

$$P(x_{kp} - x_{ip} > 0) = 1 - NCDF\left(\frac{\tilde{x}_{ip} - \tilde{x}_{kp}}{\sqrt{2} \cdot \sigma}\right)$$

or

$$P(x_{kp} - x_{ip} > 0) = NCDF\left(\frac{\tilde{x}_{kp} - \tilde{x}_{ip}}{\sqrt{2} \cdot \sigma}\right)$$

Thus, the probabilistic comparison of two actual criterion values, given that the noise model is Gaussian in nature can be expressed in terms of the difference of their measured values, and the parameter σ , by using the CDF for the standard normal $N_x(0, 1)$ to get the corresponding value of the probability.

Equation 6.10 expresses the probability that the actual value, x_{kp} is greater than

¹⁴Since we are subtracting the mean value and dividing by the standard deviation

x_{ip} by making use of corresponding measured values, \tilde{x}_{kp} and \tilde{x}_{ip} , on a particular criterion p , along with the M-function. In the description of the choice filter based on Bayesian decision theory, we stated that a single expression of conservatism μ from the user can be used to produce an appropriate survivor set from the Bayesian ordering. We show for the TBDFs that a similar quantity μ can be elicited and translated to an appropriate set of tolerances permitting the application of TBDFs to solve the problem.

Consider a pair of alternatives C_i and C_k . In order to infer that C_k dominates C_i we want the following to hold in terms of the actual criteria values taken by these two alternatives:

$$\exists p (x_{kp} > x_{ip}) \wedge \forall q (x_{kq} \geq x_{iq}) \quad (6.13)$$

The TBDFs replace the comparisons between actual criterion values in the above rule by comparisons between the corresponding measured criterion values along with suitably chosen tolerance values inside the clauses. Consider a comparison between two measured criterion values of the form below,

$$\tilde{x}_{kp} > \tilde{x}_{ip} + \epsilon_p$$

which is the same as,

$$\tilde{x}_{kp} - \tilde{x}_{ip} > \epsilon_p$$

or,

$$M(\tilde{x}_{kp} - \tilde{x}_{ip}) > M(\epsilon_p).$$

Since ϵ_p is the difference between the measured criteria values, it is the same as δ in Equation 6.10. Therefore the above equation along with Equation 6.10 allows us to infer that,

$$\tilde{x}_{kp} > \tilde{x}_{ip} + \epsilon_p \iff P(x_{kp} > x_{ip}) > \mu \quad (6.14)$$

where,

$$\mu = M(\delta) = a \text{ constant}$$

Equation 6.14 shows the relation between comparing two actual criterion values and comparing the corresponding measured values with a suitable tolerance introduced

in the comparison. Thus, one way to interpret the tolerances in the TBDF rules for decision problems with uncertainty is in the form of expressions of probabilistic assurances desired by the user. In other words, since the tolerances and the measured criterion values can be used to compute the probability that the corresponding actual values stand in a certain order relation, the user's expression of how large a probabilistic assurance he desires can be used to infer the tolerances to be used in a tolerance-based dominance rule. So, if an expression of the quantity μ can be elicited from the user, then this expression can be used to infer the corresponding tolerances for each of the criteria as,

$$\epsilon_p = M^{-1}(\mu), \quad (6.15)$$

and used in the TBDF decision rule. The user can express this quantity in a scale of 0 to 1. If the user can express the value of μ then this value can be used to infer the tolerances for all the criteria.

6.5.2 Bounding assurances based on expressions of conservatism

In the previous section, we discussed how a single user expression of conservatism(μ) can be used to produce tolerance values for the criteria. These tolerance values can be considered as being inferred from the bounds being placed on the unbounded noise models for the various criteria. Using these tolerances, the nearly-sufficient set and the nearly-necessary set can be produced by using the Superstrict and the Strongly-Strict filter as discussed in Section 6.4. However, and as mentioned earlier, since there is a non-zero probability that the actual values could lie outside the bounded range, the assurances on the nearly-sufficient and the nearly-necessary sets are no longer complete. In other words, there is a non-zero probability that the nearly-sufficient set leaves out a true positive and that the nearly-necessary set contains a false positive. In the following, we derive bounds related to the above two quantities. We first derive an upper bound on the probability that an alternative not in the nearly-sufficient set is optimal. Then we derive an upper bound for the probability *that any alternative in the nearly-necessary set is dominated by a single other alternative* (pairwise optimality

assurance for the nearly-necessary set). This quantity is related to, but not the same as the probability of there being a suboptimal alternative in the nearly-necessary set.

Bounds for the nearly-Sufficient set

Suppose that $C_i \notin S(Ss(\epsilon), \tilde{\mathcal{X}})$. In other words, C_i does not belong to the nearly-sufficient set. We are interested in the probability that C_i is optimal. This is the same as the probability that C_i is not dominated by any of the other alternatives, which is the same as the probability that C_i is better on some criterion or the other with respect to each of the other alternatives. In other words,

$$C_i \text{ is optimal} \iff \forall j \exists k (x_{ik} > x_{jk}), j \neq i$$

$$P(C_i \text{ is optimal}) = \prod_{j=1}^n P(\bigcup_{k=1}^m (x_{ik} > x_{jk}))$$

Since the product is over all n alternatives, this product contains terms related to alternatives which are in the nearly-sufficient set. Let $C_j \in S(Ss(\epsilon), \tilde{\mathcal{X}})$ be one such alternative. Then we have,

$$P(C_i \text{ is optimal}) < P(\bigcup_{k=1}^m (x_{ik} > x_{jk})) \quad (6.16)$$

The right hand term is the probability that C_i is better than C_j in at least one criterion which can be written as:

$$P(\bigcup_{k=1}^m (x_{ik} > x_{jk})) = 1 - \prod_{k=1}^m P(x_{jk} \geq x_{ik}) \quad (6.17)$$

Now since $C_i \notin S(Ss(\epsilon), \tilde{\mathcal{X}})$ and since $C_j \in S(Ss(\epsilon), \tilde{\mathcal{X}})$, we have,

$$\forall q \tilde{x}_{jq} \geq \tilde{x}_{iq} + \epsilon_q \wedge \exists r \tilde{x}_{jq} > \tilde{x}_{iq} + \epsilon_r.$$

This implies that,

$$\forall q P(x_{jq} \geq x_{iq}) \geq \mu \wedge \exists r P(x_{jq} > x_{iq}) > \mu. \quad (6.18)$$

From Equations 6.17 and 6.18, we see that

$$P\left(\bigcup_{k=1}^m x_{ik} > x_{jk}\right) < 1 - \mu^m.$$

From Equation 6.16

$$P[\text{An alternative excluded by the nearly-Sufficient Set is optimal}] < 1 - \mu^m \quad (6.19)$$

Bounds for the nearly-Necessary set

Suppose that $C_i \in S(Sts(\epsilon), \tilde{\mathcal{X}})$. In other words, C_i belongs to the nearly-necessary set. We are interested in the probability that C_i is dominated by some other alternative, say C_j . So, we are interested in:

$$P(C_i \text{ is dominated by } C_j) = \prod_{r=1}^m P(x_{jr} \geq x_{ir}) [1 - \prod_{s=1}^m P(x_{is} = x_{js})] \quad (6.20)$$

Now since C_i survives the Strongly-Strict filter we know that C_i is ϵ -better than every other alternative on some criterion. Thus,

$$\exists q \tilde{x}_{iq} > \tilde{x}_{jq} + \epsilon.$$

From Equation 6.14, this implies that

$$\exists q P(x_{iq} > x_{jq}) > \mu.$$

This implies that

$$\begin{aligned} \exists q 1 - P(x_{jq} \geq x_{iq}) &> \mu. \\ \exists q P(x_{jq} \geq x_{iq}) &< 1 - \mu. \end{aligned} \quad (6.21)$$

From Equations 6.20 and 6.21, we see that,

$$P(C_i \text{ is dominated by } C_j) < 1 - \mu.$$

Or,

$$P[\text{An alternative included in nearly - Necessary Set is pairwise dominated by some alternative}] < 1 - \mu \quad (6.22)$$

Equations 6.19 and 6.22 provide different sorts of assurances related to the nearly-sufficient set and the nearly-necessary set respectively as a function of μ . Indeed there is no need for the μ based on which the nearly-necessary set is generated to be the same as the μ for the nearly-sufficient set. Hence, if we label the μ 's differently, μ_n for the nearly-necessary set and μ_p for the nearly-sufficient set, then rewriting Equations 6.19 and 6.22 we get:

$$P(C_i \text{ not in the nearly-sufficient set is suboptimal}) > [\mu_p]^m, \text{ and} \\ P(C_i \text{ in the nearly-necessary set survives any pairwise comparison}) > \mu_n$$

Now, the above μ expressions can be considered as expressions of the user related to the above probabilities. In other words, the user could be asked to express the assurance that he desires related to the following probabilities:

- the minimum desired probability that an alternative not in the nearly-sufficient set is suboptimal (i.e., $[\mu_p]^m$) and,
- the minimum desired probability that an alternative in the nearly-necessary set survives any pairwise comparison (i.e. μ_n).

Each of these two μ 's can be used to produce two different sets of ϵ values for the criteria. The nearly-sufficient set is then created by using the Superstrict-dominance filter and the nearly-necessary set by using the Strongly Strict filter. Unfortunately, we see that these bounds are weak and therefore not useful operationally.

The analysis, however, allows us to verify a few intuitions. Since the tolerances are directly proportional to the value of μ , we can see that when a user desires a very high assurance that each alternative in the nearly-necessary set is not suboptimal, the equivalent value of μ_n is quite high and as a result, the resulting tolerances for the criteria increase in magnitude as well. This results in the production of a subset

of the original set and therefore the size of the nearly-necessary set reduces in order to accommodate the extra assurance required by the user. Indeed setting the value of $\mu_n = 1.0$ results in the nearly-necessary set being empty (proven as Theorem 3.2 for the Strongly Strict filter).

Similarly, a high assurance on the probability related to the event that the nearly-sufficient set does not exclude a true positive relates to a large value of μ_p which in turn increases the values of the criteria tolerances and thus produces a larger nearly-sufficient set. Also, the extreme case wherein the user demands can only be met by setting $\mu_p = 1.0$, the tolerances result in all of the alternatives being in the nearly-sufficient set (these have been proven in Chapter 3 as Theorem 3.3 for the Superstrict filter).

6.5.3 Generalized nearly-Necessary and nearly-Sufficient sets: The In-Between Sets

From Equation 3.2 in Chapter 3,

For $\tilde{\mathcal{X}} = \{C, \tilde{X}\}$

$$\begin{aligned} \emptyset &= S(Sts(\alpha_1)) \subseteq S(Sts(\alpha_2)) \subseteq S(Sts(\alpha_3)) \cdots \\ &\subseteq S(Sts(\alpha_p)) = S(D) = S(SS(\omega_q)) \\ &\subseteq S(SS(\omega_5)) \subseteq S(SS(\omega_6)) \cdots \subseteq S(SS(\omega_r)) = C. \end{aligned} \quad (6.23)$$

where $\alpha_1 > \alpha_2 > \alpha_3 \cdots > \alpha_p \geq 0 \leq \omega_q < \omega_5 < \omega_6 \cdots < \omega_r$.

Equation 3.2 shows that there is a spectrum of survivor sets starting from the null set at the left end to the entire set of alternatives at the right end in the above family. The survivor set represented by $S(D)$ contains the measured dominance survivors. All survivor sets to the left of this term are subsets of the measured dominance set; they are obtained by the application of the Strongly Strict filter, and are therefore *nearly-necessary sets* parameterized on the tolerances α_i . Similarly, all the survivor sets which contain the measured dominance survivors are *nearly-sufficient sets* obtained by the application of the Superstrict filter, parameterized over tolerances ω_i .

The above equation also shows that there is a tolerance vector for which the application of the Superstrict filter produces the measured dominance survivor set and the same is true for the Strongly strict filter. This is represented by the equality in the second line of Equation 3.2. This imparts an interesting property to the measured dominance survivors - it is the smallest *nearly-sufficient set* that can be expected to contain *all* of the true positives. Conversely, it is also the largest *nearly-necessary set* that can be expected to contain *only* true positives. In other words, the measured dominance survivors represent the in-between set which best compromises necessity, sufficiency and pruning at the same time.

There is another manner in which measured dominance, nearly-necessary sets and nearly-sufficient sets can be related and this is shown in Figure 12. The figure shows an alternative C_i plotted as a point on a 2-D axes for a 2-criteria problem. The tolerance vector for the problem is denoted by $\epsilon = \{\epsilon_1, \epsilon_2\}$. The figure shows a 2ϵ box around the point being considered, with the point at the center of the box. In the figure, the region bounded by the angle labeled BB1 in its first quadrant represents the region in which some alternative will have to lie in order for C_i to not belong to the measured dominance set. The angles AA1 and CC1 bound similar regions in order for the alternative C_i to not lie in the nearly-necessary set and the nearly-sufficient set for tolerance ϵ respectively. This also shows the inclusion-relation between the three sets. In other words, if an alternative C_j lies in the first quadrant of CC1, then this would imply that C_i cannot belong to the nearly-sufficient set. Because C_j lies in the first quadrant of CC1, it will also lie in the first quadrants of BB1 and AA1 as well, thereby preventing C_i from lying in the measured dominance set or the nearly-necessary set as well. Therefore, as we inferred previously, if C does not belong to the nearly-sufficient set, then it cannot belong to the measured dominance set or the nearly-necessary set.

From the figure we can interpret the act of declaring an alternative as belonging to the nearly-necessary set or to the nearly-sufficient set, as one of weakening or strengthening the conditions to test for dominance. More specifically, if we consider an alternative C_k at the junction point of CC1, then the first quadrant of CC1 represents the region which controls whether or not C_k will lie on the dominance set much like BB1 does for the alternative C_i . Now we can consider C_k to be a point

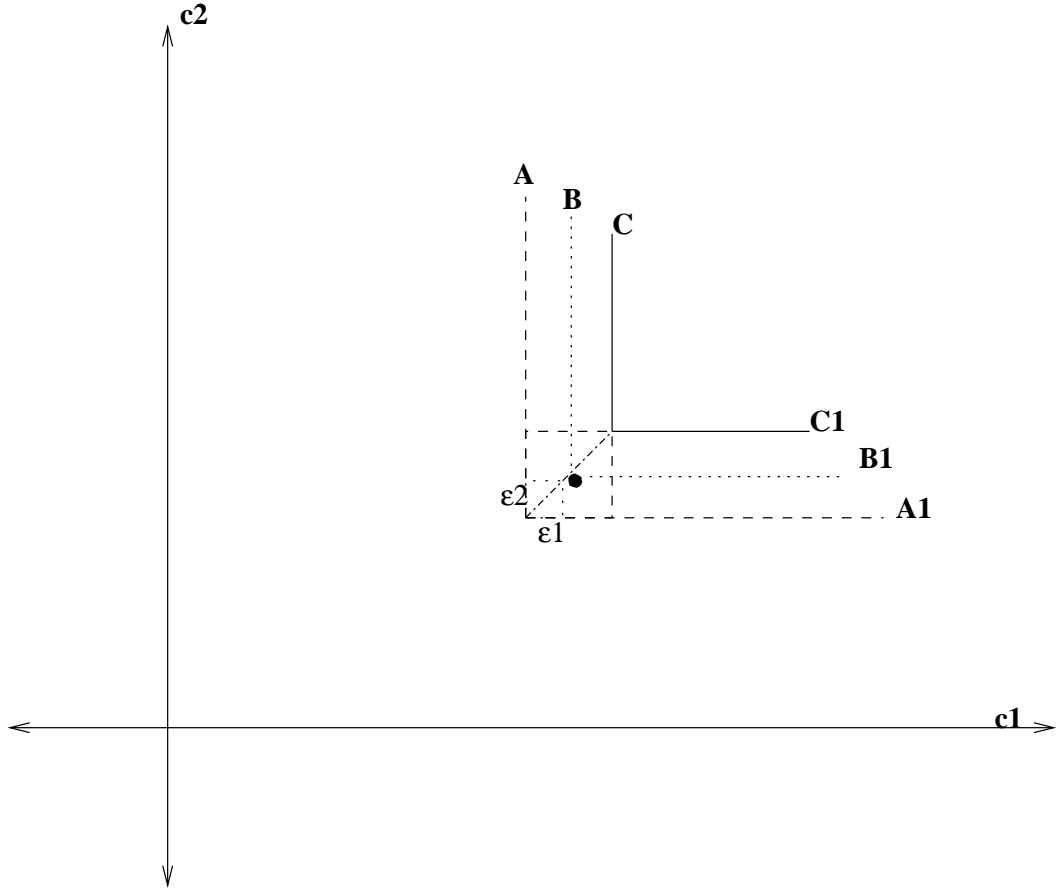


Figure 12: Relating nearly-necessary (ϵ) and nearly-sufficient(ϵ) sets to measured dominance

obtained from C_i but with its criteria values increased by ϵ each. In other words, testing whether C_i belongs to the nearly-sufficient set is equivalent to testing if an ϵ -upgraded version of C_i will survive dominance with respect to the original set of alternatives. In other words, testing for *sufficiency* entails strengthening the alternative and checking if it is a dominance survivor. More precisely, The following can be shown to be true for every alternative belonging to the nearly-sufficient set $PS(\epsilon)$:

Theorem 6.6 *Each alternative in the Nearly-Sufficient Set satisfies the following condition with respect to every other alternative:*

$$(\forall C_i \in PS) (\forall C_j \in C) \exists k(\tilde{x}_{ik} + \epsilon \geq \tilde{x}_{jk}) \quad (6.24)$$

Proof: See Appendix A.

On the other hand, the junction point of the angle AA1 can be seen as obtained by decreasing the criteria values of C_i by ϵ . So testing for *necessity* entails weakening the alternative (by ϵ -downgrading its criteria values) and checking if it continues to survive measured dominance even under the weakened conditions. Specifically, we know from the decision rule for the Strongly-strict filter that the following is true for each alternative in the nearly-necessary set $NS(\epsilon)$:

$$(\forall C_i \in NS) (\forall C_j \in C) \exists k(\tilde{x}_{ik} > \tilde{x}_{jk} + \epsilon). \quad (6.25)$$

The in-between sets for the given tolerance vector are represented in the figure by the diagonal of the box passing through all the three junction points and shown in the figure. The diagonal represents a continuum of points along which the alternative C_i could be either weakened or strengthened to obtain corresponding in-between sets.

6.6 Mapping user needs to near-sufficiency and near-necessity

In Section 6.5.2, we saw that although we could map the assurances desired by the DM on the nearly-necessary and nearly-sufficient sets to appropriate values of tolerances, the bounds based on which this mapping was done were too weak to produce operationally useful survivor sets. In this section, we show that there is another manner in which user expressions of a different kind can be mapped onto tolerance values that can be used to produce an appropriate survivor set.

The optimal set can be produced by applying the following decision rule to the actual criteria values:

$$\text{Retain } C_i \iff \forall j \exists k(x_{ik} > x_{jk})$$

This is logically equivalent to the dominance rule but for our purposes this characterization is more useful. Now, in the presence of uncertainty, a similar decision rule expressed probabilistically can be of the following form:

$$\text{Retain } C_i \iff \forall j \exists k (P(x_{ik} > x_{jk}) > \mu). \quad (6.26)$$

where μ is the probabilistic assurance that the user requires related to the event that each survivor is better than every other alternative on some criterion. If the user is able to express μ , then according to Equation 6.14, we know that this decision rule translates to the following one:

$$\text{Retain } C_i \iff \forall j \exists k (\tilde{x}_{ik} > \tilde{x}_{jk} + \epsilon).$$

The interesting result of applying this decision rule based on the μ expression of the user is that, for the entire range of μ (i.e. $[0,1]$), it is now possible to deploy a corresponding TBDF to produce the survivor set according to the decision rule above. This is because it can be shown that if μ is greater than 0.5, then the corresponding value of ϵ above will be positive. From Equation 6.25, we know that the survivor set satisfying the above requirement can be obtained by applying $Sts(\epsilon)$. On the other hand if the μ expressed by the user is less than 0.5, then the above requirement on each survivor can be rewritten as

$$\forall j \exists k (\tilde{x}_{ik} + \epsilon > \tilde{x}_{jk}).$$

which according to Theorem 6.6 can be obtained by applying $SS(\epsilon)$. And if μ is equal to 0.5, the corresponding value of ϵ becomes 0 and the decision rule becomes the rule for measured dominance. In other words, every survivor set that satisfies Equation 6.26 pertaining to some μ value expressed by the user can be obtained by the application of an appropriate choice filter. We already saw this mapping earlier, as expressed in Equation 6.11.

Also, the pairwise probability of survival of an alternative, C_i , with respect to

another alternative say C_j is expressed as:

$$P(E_i, C_j) = 1 - \prod_{k=1}^m P(x_{jk} > x_{ik}). \quad (6.27)$$

Since every survivor obtained by the use of either the Strongly-strict or the Super-strict filter satisfies Equation 6.26, we must have for any survivor C_i and with respect to any other alternative C_j ,

$$P(E_i, C_j) > \mu.$$

This indicates that the pairwise probability of survival of each survivor produced by the above probabilistic rule will be greater than μ . Hence, the user expression of μ can also be obtained in terms of the degree of assurance he needs related to the pairwise survival probability of each survivor with respect to any of the remaining alternatives. This value of μ can be translated to an equivalent value of tolerance and either the Strongly-Strict, measured dominance, or Superstrict filter can be applied to produce the survivor set, depending upon whether μ is greater than, equal to, or less than 0.5 respectively.

6.7 Relating In-between sets to Pruning demands

Figure 10 shows that for a bounded noise model, the in-between sets are sets which contain the necessary set and which are subsets of the sufficient set. In the case of an unbounded, tolerance model however, there is no predefined notion of a necessary and a sufficient set. In fact, as we discussed in the previous section, different degrees of necessity and sufficiency can be applied by using different values of tolerances. As discussed in the previous section the user's μ expressions can be used to produce the nearly-necessary and nearly-sufficient sets according to those expressions. Thus the application of this technique is contingent upon the user being able to express the μ values, and upon there being sufficient information to allow the implementation of the $M()$ function, to obtain the equivalent tolerance values.

For the class of users who can only express their concerns in terms of their pruning requirements, it is not clear how the nearly-necessary and nearly-sufficient sets can be produced based on the pruning requirements, unless these requirements can be

mapped somehow to μ values. Even if the user is able to express μ values, he might not be satisfied with the results because they might not be in accordance to his pruning requirements¹⁵. Thus, it becomes necessary to extend the technique of producing nearly-necessary and nearly-sufficient sets, to include the pruning requirements of the user as well. In this section, we first show how the idea of nearly-necessary and nearly-sufficient sets applies even in the absence of the μ expressions. We also show how pruning demands of the user can be incorporated in the technique.

Consider an unbounded noise model for which we do not have the μ expressions. However, we do have knowledge about the deviations or the σ 's on the various criteria. We could use these deviations to calibrate the tolerances along the criteria. For example consider three alternatives C_i , C_j , and C_k for an $m = 2$ problem with the measured values as indicated below:

	1	2
C_i	15	4
C_j	20	1
C_k	5	3

Based on the measured values alone, it is not clear which of the above three alternatives have the highest chance of being truly optimal. For example, in comparing the alternatives C_j and C_k , we see that C_j is 15 units superior than C_k on criterion 1, while C_k is 2 units better than C_j in criterion 2. Of course, since the units along the criteria are incommensurable, there is no way to know which of these *superiorities* to trust more. Without an idea of the range of actual values that these measured values can be expected to take, it is both possible to imagine scenarios where C_j has a higher chance of being optimal relative to C_j and conversely other scenarios wherein C_k has a higher chance of being truly optimal. If however, we have additional information about the noise models for each of these 2 criteria, then we will have more information based on which to assign credibilities to the above measured values, and thereby infer which has the higher chance of being truly optimal.

More specifically, suppose that $\sigma_1 = 5$ and $\sigma_2 = 1$. If we now normalize the values on each criterion r according to the following technique:

¹⁵As discussed, the nearly-necessary set could be empty and the nearly-sufficient set as large as the entire set of alternatives, if the μ expressions are too conservative

1. The smallest value on criterion r , say \tilde{x}_{sr} is set to 0.
2. The remaining values in r are expressed as units of deviations away from the smallest value. Thus, a value \tilde{x}_{ar} will be converted to $(\tilde{x}_{ar} - \tilde{x}_{sr})/\sigma_r$.

The result of the normalization is that a difference of k units between the normalized values for any single criterion represents a difference of k deviations along that criterion. For the above problem, the normalized values obtained according to the procedure described and the given values of deviations for the criteria is as below:

	1	2
C_i	2	3
C_j	3	0
C_k	0	2

From the above, we see that normalized values for \tilde{x}_{j1} and \tilde{x}_{k1} are 3 deviations apart in criterion 1, as are the values for \tilde{x}_{i2} and \tilde{x}_{j2} along criterion 2. Thus we have,

$$\frac{\tilde{x}_{j1} - \tilde{x}_{k1}}{\sigma_1} = \frac{\tilde{x}_{i2} - \tilde{x}_{j2}}{\sigma_2}.$$

For the Gaussian noise model this translates to,

$$G\left(\frac{\tilde{x}_{j1} - \tilde{x}_{k1}}{\sqrt{2}\sigma_1}\right) = G\left(\frac{\tilde{x}_{i2} - \tilde{x}_{j2}}{\sqrt{2}\sigma_2}\right).$$

From Equation 6.8 therefore,

$$P(x_{j1} > x_{ik}) = P(x_{i2} > x_{j2}).$$

In other words, the above normalization standardizes the measured values across all criteria so that a unit difference between two normalized values along all criteria can be considered as equivalent. This restricts us to set the tolerance values for all criteria to be the same (i.e., or tolerance vectors which have identical tolerance values for all criteria).

With this requirement on the tolerance vector, we see that the following are true:

- By considering all possible values that can be assigned to the tolerance vectors, the only nearly-necessary sets that can result by the application of the Strongly Strict filter using these tolerance vectors are: $\{C_i\}$, and $\{C_i, C_j\}$.
- By considering all possible values that can be assigned to the tolerance vectors, the only nearly-sufficient sets that can result by the application of the Super-strict filter using these tolerance vectors are: $\{C_i, C_j\}$, and $\{C_i, C_j, C_k\}$. In fact we already know that all nearly-sufficient sets contain the measured dominance set ($\{C_i, C_j\}$ in this case).

In other words, once we are restricted to the use of only tolerance vectors with equal components, there is no way to get nearly-necessary sets like $\{C_j\}$ or $\{C_j, C_k\}$ or nearly-sufficient sets like $\{C_k\}$ based on the normalized values. Another way to interpret this result is that we can consider the three alternatives to be ordered as $\{C_i, C_j, C_k\}$ where only the subsets containing the topmost contiguous set of alternatives up to the measured dominance set can form nearly-necessary sets; all other subsets are eliminated from being nearly-necessary sets (i.e., there is no tolerance vector for which they can form a nearly-necessary set). Similarly, only the subsets containing all of the measured dominance alternatives appended with contiguous alternatives from the non-survivors of measured dominance can form nearly-sufficient sets. Moreover, we can index each alternative with a minimum tolerance vector at and above which the alternative will cease to be a member of the nearly-necessary/nearly-sufficient set.

For the above problem these values are $\{2, 1, 1\}$ for C_i , C_j and C_k respectively. In other words, tolerance vectors less than 2 but greater than 1 produce $\{C_i\}$ as the nearly-necessary sets, tolerance vectors with values less than 1 but greater than 0 produce $\{C_i, C_j\}$ as the nearly-necessary set and being the measured dominance set, it is the largest nearly-necessary set. This measured dominance set is also the smallest *nearly-sufficient set*. And finally for tolerance values greater than 1, we get the entire set of alternatives as the nearly-sufficient set. This ordering now represents an ordering of the different survivor sets from Equation 3.2 starting from the null set which can be considered as the top of the ordering, any in-between set being a top, contiguous set of alternatives from the ordering, with sets greater bigger than, and

therefore including, the measured dominance set being nearly-sufficient sets while sets which are subsets of the measured dominance set being nearly-necessary sets. One way for the user to choose from this ordering is to impose a pruning expression either in terms of how many alternatives he expects, or individual pruning expressions for nearly-necessary and nearly-sufficient sets.

The question is whether such an ordering can be efficiently computed for the general case with n alternatives. More specifically we desire an ordering such that, if one selects the topmost r alternatives, indicated as the set I , then the following two propositions must hold for any such set I :

- $\forall C_i \in I, \forall C_j \in X - I$, there is no tolerance vector for which C_j belongs to the corresponding nearly-necessary set and C_i does not. Conversely there is a tolerance vector for which C_i belongs to the corresponding nearly-necessary set and C_j does not.
- $\forall C_i \in I, \forall C_j \in X - I$, there is no tolerance vector for which C_j belongs to the corresponding nearly-sufficient set and C_i does not. Conversely there is a tolerance vector for which C_i belongs to the corresponding nearly-sufficient set and C_j does not.

Using this ordering, the user can select any set I either according to his pruning demands or if he has expressions for the nearly-necessary and nearly-sufficient sets of interest to him in the form of the μ values. We next describe the procedure by which this total ordering is computed for n alternatives.

6.7.1 A procedure to produce an ordering of In-between sets

Given a set X of n alternatives each with m measured criteria values, deviations σ_i for the various criteria and assuming the noise models to be Gaussian:

1. For each criterion i , normalize the individual values as described above (set the minimum along the criterion to zero, the remaining values are standardized by dividing their difference from the minimum value with the σ_i).
2. For each $C_j \in X$,

3. For every other $C_k \in X$,
 - (a) Compute the vectors $v_{jk} = \max[X_j - X_k]$ and $v_{kj} = \max[X_k - X_j]$
 - (b) if $v_{jk} > v_{kj}$ $tolmat(j, k) = v_{kj}$
 if C_j dominates C_k
 $tolmat(k, j) = v_{jk}$
 else if C_j and C_k are tied
 $tolmat(k, j) = \infty$
 - (c) else $tolmat(k, j) = v_{jk}$
 if C_k dominates C_j
 $tolmat(j, k) = v_{kj}$
 else if C_j and C_k are tied
 $tolmat(j, k) = \infty$
4. $tolerances[1..n] = \min[tolmat(:, 1..n)]$
5. $order = getindex(sort(tolerances(1..n)))$

Step 1 normalizes all the criteria values according using the σ values for the individual criteria by the technique described above. The remaining steps compute n values for each of the n alternatives, thereby creating entries for an $n \times n$ matrix. This matrix is shown in Figure 13. In case alternatives C_i and C_j are tied according to measured dominance then each entry ij in the matrix corresponds to the value of tolerance for which the alternative C_i will be strong-strictly dominated by alternative C_j . If there is such a tolerance vector, then there is no vector for which C_j will be strong-strictly dominated by C_i and therefore the corresponding entry ji is set to ∞ . All diagonal entries are set to ∞ to indicate the there is no tolerance vector for which an alternative can strong-strictly dominate itself.

If, on the other hand, C_i and C_j happen to be such that one measured dominates the other, then the row entry for the dominator is set to the value at which the dominated alternative becomes ϵ -indistinguishable in all criteria with respect to it. The row entry for the dominated alternative is set to the maximum value of the tolerance vector at which it becomes ϵ indistinguishable with its dominator in at

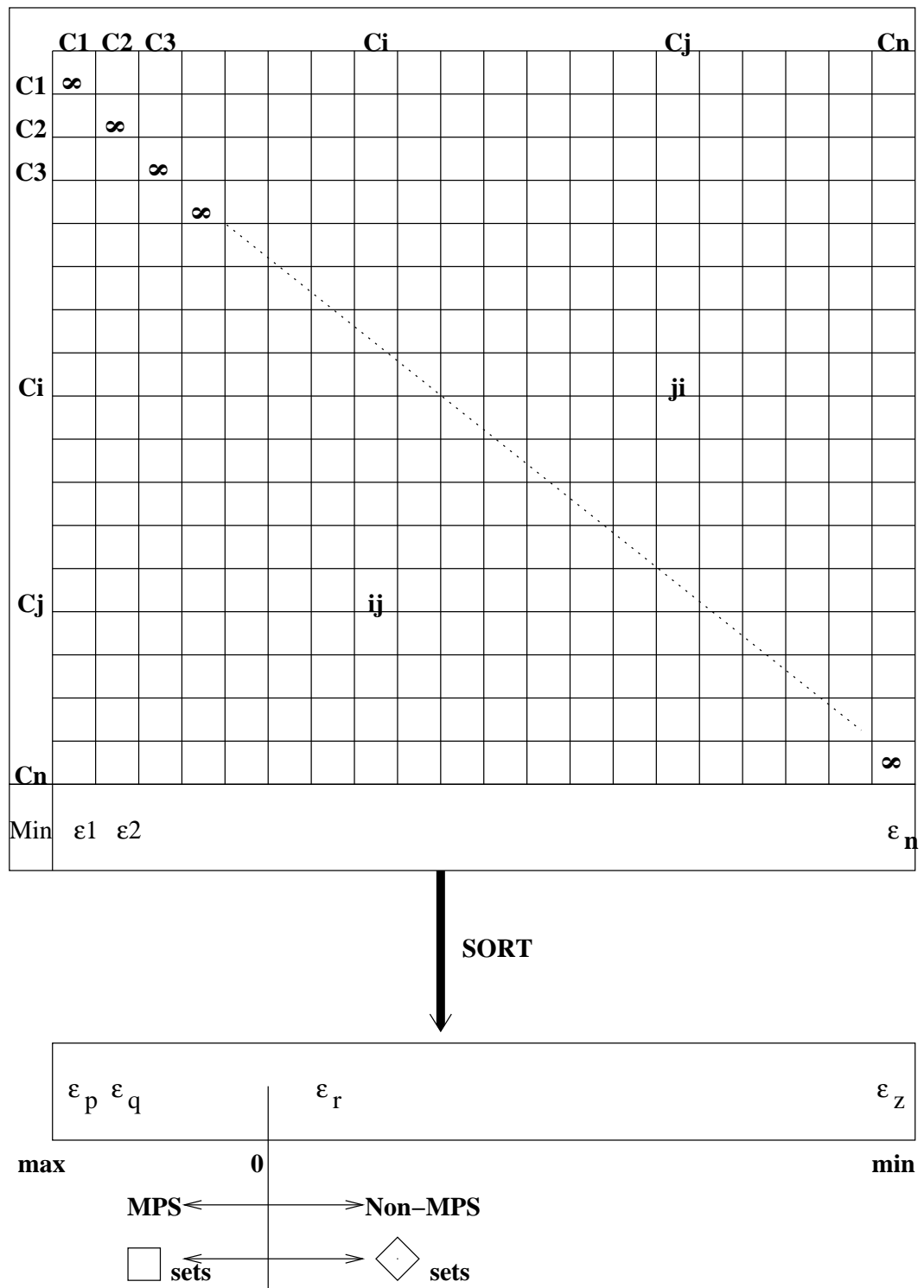


Figure 13: An ordering to produce in-between sets

least one criterion, thereby making it the minimum tolerance vector at which the dominated alternative is inserted into the sufficient set.

Once all of the matrix entries are filled, Step 4 computes the minimum of all row-entries for each alternative. This will produce a vector of length n with one tolerance value for each alternative. The tolerance value for any alternative C_k from this vector indicates the smallest value of tolerance vector for which C_k will be either strong-strictly dominated (in case C_k is a measured dominance survivor), or superstrictly-dominated (in case C_k is not a measured dominance survivor) by some alternative in the set. Next, the alternatives are ordered by sorting according this tolerance vector. This ordering is shown at the bottom of Figure 13. The above algorithm requires the computation of n^2 entries of the matrix and is therefore $O(n^2)$. However, there is an alternative technique to produce the above ordering which is more tractable and will be discussed in the next section.

In this ordering of alternatives, any pruning expression of the user can be met by selecting the appropriate number of alternatives from the ordering. In fact, for any such in-between set, the corresponding tolerance value indexing its lower most alternative can be used to compute μ values and provide assurances of the kind derived in the previous section for the in-between set. Conversely, if the user expresses μ values, then the corresponding nearly-necessary and nearly-sufficient sets can be picked from the above ordering by using the μ values.

6.7.2 Reducing the computational complexity of producing the ordering of in-between sets

In this section, we describe a modification to the previously described technique of producing the ordering of in-between sets. The modification will reduce the overall complexity of the algorithm from $O(n^2)$ to $O(n \log^{m-1} n)$. From Equation 3.2, we see that the above ordering will always contain the measured dominance survivors as an in-between set. Also, the nearly-sufficient sets pertain to the application of the Superstrict filter which entails adding new alternatives to the existing measured dominance set. Theorem 6.6 indicates the test condition for something to belong to the nearly-sufficient set $PS(\epsilon)$. According to the equation, it is necessary that the

condition hold for each $C_i \in PS(\epsilon)$ with respect to each of the other $n-1$ alternatives. However, and fortunately, it suffices to see if the condition holds with respect to the measured dominance alternatives alone. To see this, suppose alternative C_i passes the test with respect to a single measured dominance alternative C_j . This implies that C_i will pass the test according to all alternatives which are dominated by C_j . This implies that if the alternative passes the test condition according to all the measured dominance alternatives, then it has to pass the test condition with respect to every non-survivor of measured dominance. All of this implies that if one can ascertain for some alternative C_i that it satisfies the condition with respect to the measured dominance set, then it will satisfy the condition for all of the alternatives. Therefore, for the portion of the ordering below the measured dominance set, only $n - |S(D, \tilde{\mathcal{X}})| \cdot |S(D, \tilde{\mathcal{X}})|$ entries need to be computed instead of $n - |S(D, \tilde{\mathcal{X}})| \cdot n$.

For the nearly-necessary sets, however, we need to check for the test condition with respect to each of the remaining $n - 1$ alternatives. This is because a given measured dominance survivor might satisfy the test condition with respect to every other measured dominance survivor and yet there might be a dominated alternative that is ϵ -indistinguishable with respect to it in all criteria, thereby not allowing for the test condition to hold. Hence, for all measured dominance survivors, it is required that there be $n - 1$ comparisons to test the condition for belonging to the nearly-necessary set.

The overall entries which need to be computed, therefore, is as shown in Figure 14. For the measured dominance survivors labeled as the set $\{C_1, ..C_s\}$ in the figure, all n entries need to be computed. However for the alternatives/rows below C_s , which is the set of non-survivors of measured dominance, only the entries shown as boxes need to be computed. The reduction in complexity is seen by comparing Figure 14 to Figure 13; it is also indicated by the non-boxed region of the matrix in Figure 14. More precisely, the number of entries computed by this way of producing the ordering, including the $n \cdot \log(n)$ used in sorting is:

$$\begin{aligned} O(n|S(D, \tilde{\mathcal{X}})| + (n - |S(D, \tilde{\mathcal{X}})|)(|S(D, \tilde{\mathcal{X}})|) + n\log(n)) &= O(n|S(D, \tilde{\mathcal{X}})| + n\log(n)) \\ &= O(n\log^{m-1}(n)) \end{aligned} \quad (6.28)$$

since the number of dominance survivors is expected to be much smaller than n in the average case. Thus the complexity is the same as that of computing the Pareto set which is a lot more tractable than the $O(n^2)$ implementation discussed previously. In other words, given a set of alternatives, they can be ordered according to the ordering principle without a great amount of computational complexity. In-between sets can now be chosen by the user according to his pruning desires or by the μ expressions.

6.8 Performance of Choice Filters under Uncertainty

In this section we analyze all of the presented choice filters in terms of their performance in the presence of uncertainty. As mentioned previously, in the presence of uncertainty, one way to express the performance of a filter is in terms of a plot of the expected number of false positives versus the expected number of false negatives or the *FP-FN curve*, to represent the misclassification rate of a filter. The computational complexity of applying a filter is also a primary performance measure of the filter. In case the decision-maker expresses preferences related to pruning, it becomes necessary to cater to these pruning demands while simultaneously minimizing the misclassification rate. This ensures that the pruning is also directed towards producing a set which has a high probability of containing only the optimal alternatives. In other words, if the decision-maker desires additional alternatives, this must be achieved by trying to bring in the false negatives instead of the true negatives from the set of non-survivors. Similarly if additional pruning is desired, we want to eliminate false positives from the current set of survivors rather than the true positives.

From Equation 3.2 one can see how by continuously varying the value of the tolerance, one can gradually increase the survivor set starting from the null set going all the way to a value of tolerance where all the alternatives survive. Since these in-between sets are related by the set-inclusion relation, it can be said that the expected number of false negatives will reduce and that of false positives will increase as we go from smaller sets to larger sets within this spectrum. Therefore, any FP-FN tradeoffs of the user can be potentially met simply by varying the value of the tolerances.

It has already been shown how the user's FP-FN expressions in the form of the μ expressions can be met with by the choice of an appropriate ϵ vector derived from the μ

expressions. Using the μ expressions, not only can the corresponding nearly-necessary and nearly-sufficient sets be produced but assurances related to the chances that the nearly-necessary set might contain a suboptimal alternative, or that the nearly-sufficient set might have excluded an optimal alternative can be given. Additionally, pruning expressions of the decision-maker can also be met by making use of the ordering technique to produce in-between sets.

However, it is quite possible that in certain decision situations, the decision-maker might be concerned with the number of false positives or false negatives produced by a choice filter. For example, the user might be interested in how many optimal alternatives are not present in the nearly-necessary set or on how many suboptimal alternatives are included in the nearly-sufficient set. If the survivor set is produced to respond to a pruning demand of the user, then the user might be interested in knowing how many optimal alternatives are present in the survivor set, and how many optimal alternatives were not retained as survivors. In such situations, assuming that the expressions of the user can indeed be met by the use of a suitable tolerance vector and an appropriate choice filter, one would require an expression for the the expected number of false positives and false negatives produced by a choice filter.

6.8.1 Deriving an expression for the expected number of false positives and false negatives of a choice filter

In this section, an expression is derived for the expected number of false positives and false negatives produced by a choice filter $F()$. Let C_i be a randomly chosen alternative from the a given set of alternatives. By definition of a false positive we have:

$$C_i \in f_+(F, \tilde{\mathcal{X}}) \rightarrow C_i \notin S(D, \mathcal{X}) \wedge (C_i \in S(F(), \tilde{\mathcal{X}})) \quad (6.29)$$

Similarly we can express that C_i is a false negative of $F()$ by:

$$C_i \in f_-(F, \tilde{\mathcal{X}}) \rightarrow (C_i \in S(D, \mathcal{X})) \wedge C_i \notin S(F(), \tilde{\mathcal{X}}) \quad (6.30)$$

We first compute the probability that C_i is a false positive. Using Equation 6.29 we can write,

$$P(C_i \in f_+(F, \tilde{\mathcal{X}})) = P[(C_i \notin S(D, \mathcal{X})) \wedge (C_i \in S(F, \tilde{\mathcal{X}}))] \quad (6.31)$$

$$P[e1 \wedge e2]$$

The first event $e1$ occurring inside the RHS of the above equation is the event that C_i is not a Pareto survivor. Similarly, $e2$ is the event that C_i is a survivor of $F()$. The problem is that the events $e1$ and $e2$ are not independent. Hence the probability expression on the RHS of Equation 6.29 cannot be opened as the product of probabilities of the individual events. One way to view the problem is to start with a randomly chosen alternative, say C_1 . Along with the remaining $(n - 1)$ alternatives, we can consider these n points as forming a configuration. For this *fixed* configuration, we first compute the probability that C_1 will be dominated in the configuration. To this configuration, we now apply the noise model thereby corrupting the configuration in all possible ways according to the noise distribution. For each such corrupted variant of the original configuration, we next compute the probability that the given alternative in its corrupted version, \tilde{C}_1 will be a survivor according to the filter F being considered. Once this probability is computed, we will have the probability that a given configuration results in labeling C_1 as a false positive. We now, try out all possible configurations as dictated by the prior distributions, repeating all of the above for each such configuration. The final outcome will be the probability that a randomly chosen alternative is a false positive of F . By using the above technique, an expression for the probability that a randomly chosen alternative C_1 from among $n - 1$ other alternatives, $\{C_2, \dots, C_n\}$ will be a false positive of the filter F can be given by:

$$\begin{aligned}
P(C_1 \in f_+(F, \tilde{\mathcal{X}})) &= \int_{-\infty}^{\infty} \\
&\int_{-\infty}^{\infty} \overbrace{\dots}^{n-1} \int_{-\infty}^{\infty} (1 - P(\bigcap_{i=2}^n (C_1 \ S_D \ C_i))) \cdot \\
&\int_{-\infty}^{\infty} \overbrace{\dots}^n \int_{-\infty}^{\infty} [P(\bigcap_{i=2}^n (\tilde{C}_1 \ S_F \ \tilde{C}_i)) P(\tilde{C}_i | C_i)] P(\tilde{C}_1 | C_1) d\tilde{C}_1 \cdots d\tilde{C}_n \cdot \\
&P(C_2) \cdots P(C_n) dC_2 \cdots dC_n \cdot \\
&P(C_1) dC_1
\end{aligned} \tag{6.32}$$

The innermost set of integrals occurring in the third line of Equation 6.32 compute the probability that a given alternative C_1 will survive the filter F in the presence of $n - 1$ other alternatives, allowing for each of the n alternatives to take all possible measured values according to the noise model. The term $(\tilde{C}_1 \ S_F \ \tilde{C}_i)$ is the event that the alternative C_1 survives the alternative C_i according to the decision rule of the filter F when applied to their measured criteria values. In other words the term S_F is the proposition *survives according to the decision rule of filter F*. Similarly, the term $(C_1 \ S_D \ C_i)$ is the event that alternative C_1 survives alternative C_i according to the dominance rule, or filter D , applied to the actual criteria values taken by these two alternatives. Again, the term S_D is the proposition *survives according to dominance rule*. The second layer of integrals spanning lines 2 through 4 of the equation change actual configurations according to the prior distribution of the criteria values. The outermost integral makes the expression applicable for all possible choices of C_1 so the expression applies to a randomly chosen alternative.

6.8.2 A Complexity analysis of computing the expected number of false positives for a choice filter

Equation 6.32 already gives indications that the computation of the above quantity is of great complexity. In this section, we produce an order approximation on the complexity of the above computation. Again, the unit of complexity will be one pairwise operation involving either the D or the F functions. Firstly the expressions inside the

integrals are analytically complex because the two probability terms computed over the conjunction of the events involve conditional probabilities since the events are not independent. Even if we assume for simplicity that the events were independent and compute the probability as the product of the individual probabilities, these two inner products will each be $O(n^2)$ in complexity and lead to an overall $O(n^4)$ pairwise comparisons. This is just for one cycle of the integration. Assuming a granularity of k for each of the integrals in the expression, we see that the complexity becomes exponential in n . It takes k cycles on the outermost integral, and k^n on the innermost and finally k^{n-1} on the second layer of integrals. Hence the overall complexity is:

$$O(k \cdot k^{n-1} \cdot n^2 \cdot k^n \cdot n^2) = O(k^{2n} \cdot n^4)$$

Clearly, even for a small n problem the complexity of the computations is of very high order. As a result an alternative technique is required which can compute the expected number of false positives and false negatives given a filter F . One common way to tackle computations which are extremely complex is by using Monte Carlo simulations. The next section describes the technique and the experiments conducted to assess the FP-FN characteristics of the various choice filters.

6.9 Monte Carlo Simulations to assess performance of choice filters

In the previous section, the problem of computing the expected number of false positives for a choice filter F , given the set of measured criteria values for a decision problem along with the probability distributions for the actual values and for the criteria noise models, was shown to be analytically complex to solve. From the definition of a false negative, we would expect the computation of the expected number of false negatives to be equally complex. In this section, we describe one way to compare the FP-FN curves for the choice filters discussed for decision situations with uncertainty. It involves the use of Monte Carlo simulations conducted on randomly generated numbers.

6.9.1 Data Generation

For a given $n \times m$ decision problem, a matrix of $n \times m$ values are generated by using a random number generator. The distribution from which the values are generated is set to be the prior distributions for the class of problems being investigated. For our purposes we assume all prior distributions to be $N(0,1)$ or the standard normal distribution, since the normal distribution can be expected to characterize distributions to which many criteria values in the real world can be expected to belong to. This generated matrix of values is treated as the set of actual criteria values, \mathcal{X} , for the n alternatives along the m criteria.

Next, the noise model is modeled. The noise model for each criterion is taken to be $N(\mu, \sigma)$, or a normal distribution with the actual value μ as its mean and the standard deviation of σ . For a chosen noise model, and for a generated set of actual criteria values, a new matrix of $n \times m$ values is now generated by adding noise to each of the values in the original matrix according to the chosen noise model. In other words, the criterion value x_{ij} is used to generate a possible measured value \tilde{x}_{ij} using:

$$\tilde{x}_{ij} = x_{ij} + \sigma N(0, 1).$$

This newly generated matrix is treated as the set of measured criteria values, $\tilde{\mathcal{X}}$ for the decision problem.

6.9.2 Simulations

In order to obtain the performance characteristics of any choice filter $F()$, we first apply the filter to $\tilde{\mathcal{X}}$ in order to obtain the survivor set $S(F(), \tilde{\mathcal{X}})$. Since the set of actual criteria values, \mathcal{X} , is available we can find the optimal alternatives for the decision problem, namely $S(D, \mathcal{X})$, by applying the dominance filter to \mathcal{X} . Now, one can examine the survivor set produced by the filter to see how many of the optimal alternatives are retained as survivors and how many are eliminated. This will allow us to compute the number of false positives and false negatives produced by the single simulation of the decision problem. In order to obtain the expected number of false positives and false negatives (as well as true positives and true negatives), the

above simulation is repeated from start, each time beginning with a new \mathcal{X} , using the noise model to create a new $\tilde{\mathcal{X}}$, applying the filter to this newly generated set, and assessing its performance with respect to the new \mathcal{X} . If an adequate number of simulations is conducted, the Monte Carlo experiment can be expected to produce a good approximation of the true quantities of interest. Standard tables can be used to decide on the minimum number of simulations to be conducted for the results to be reliable and for our purposes, we conducted 40 simulation trials per performance measure.

6.9.3 Obtaining performance for various parameterized instantiations of a filter

Since we are interested in the performance of each of the choice filters for different values of their parameters, the above experiment will need to be conducted for each instantiated set of parameters for each filter. In other words, for each filter instantiated with a choice of tolerance vector, 40 simulation trials as described in the previous section are conducted to obtain the performance measures - expected number of false positives, false negatives, true positives and true negatives for the instantiation of the filter. In other words, each point in the FP-FN space will correspond to the FP-FN performance obtained by the application of the filter for a given tolerance vector. Once a number of such points are obtained for different values of the tolerance vector, the overall FP-FN curve is interpolated using these points.

6.9.4 Estimation of Confidence Intervals

In order to test for the statistical validity of the results, standard confidence estimates in the form of Student T-interval estimates were obtained for the the points as well. For the FP-FN characteristics, intervals for simultaneous 95% confidence regions for the pairs $\{FP, FN\}$ were obtained.

6.9.5 Results

TBDFs

The plots in Figure 15 show the results of the Monte Carlo simulations conducted for the Strongly-strict and the Superstrict dominance filters for various values of tolerances. Each plot contains two curves which show the FP-FN characteristics for the cases when $\sigma = 5\%$ and $\sigma = 20\%$ noise models respectively. As mentioned before, the curve is an interpolation of points shown on the curve. Each point corresponds to a distinct tolerance vector for the filter. The operating points for the Strongly-strict filter which obtains nearly-necessary sets are shown as *circles* in the plots while those for the Superstrict filter are shown as *stars*. The operating point where these two interpolated curves meet is the measured dominance point as is known from Equation 3.2. As proved in Theorem 6.1, all the measured dominance points lie on the line $FP = FN$ through the origin. The first plot is for an $n = 100, m = 4$ decision problem, the second one for an $n = 100, m = 6$ problem and the final one for an $n = 1000, m = 4$ decision problem. All the curves indicate how a smooth tradeoff along the FP-FN curve can be obtained by varying the tolerance vector alone. Each plot also shows that the FP-FN curve for the less noisy problem is below the one for the noisier situation. This is expected since as the noise increases, the chances that a filter will produce a false positive and false negative is bound to increase. It is important to relate each curve to Equation 3.2 in order to understand the direction of increase in the tolerance when we move from one point in the curve to the other. In other words, for the FP-FN curve related to the Strongly-Strict filter (i.e., the curve with circles as points), as we move down the curve, the tolerance vector is reducing until it hits the measured dominance point where the tolerance vector becomes zero. From this point onwards, the FP-FN curve relates to the Superstrict Filter and moving along the curve towards the right pertains to an increase in the tolerance vector.

Comparing the first two plots, we see that all else remaining the same, when the number of criteria is increased, the expected number of false negatives increases while the expected number of false positives reduces. This can be explained by considering the fact that as m increases, the probability that an alternative is optimal increases

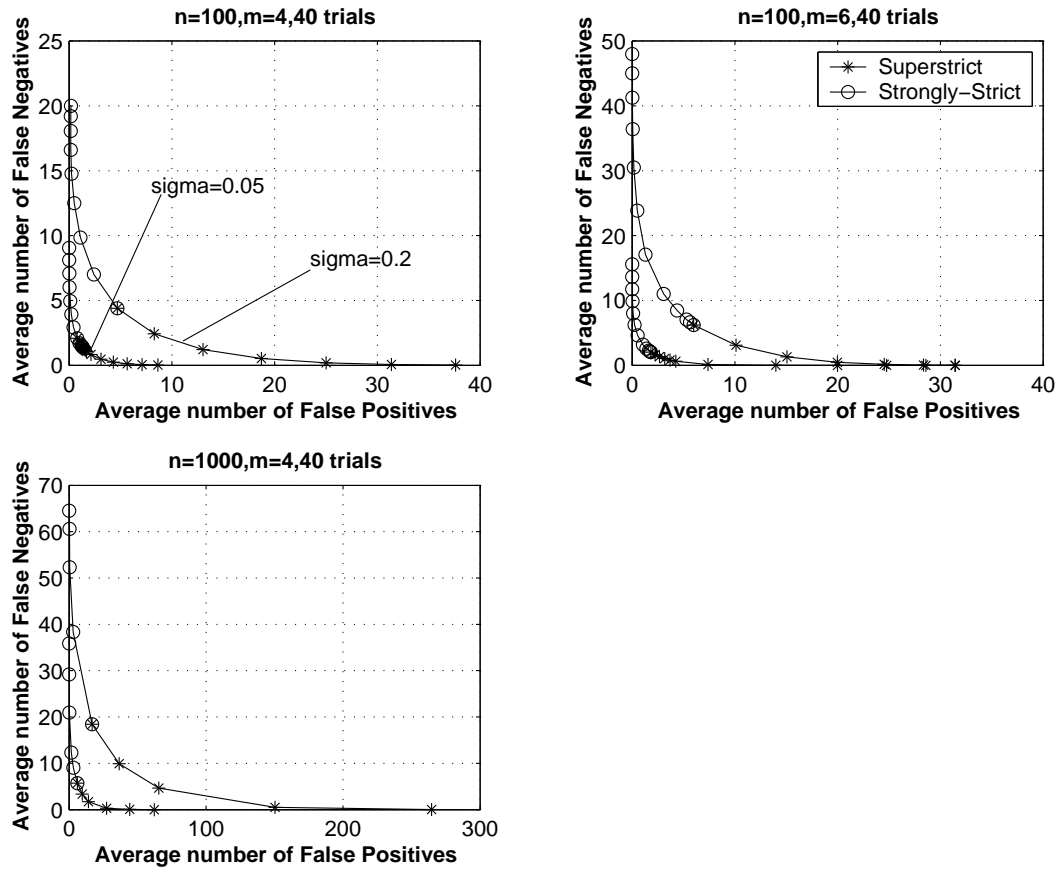


Figure 15: FP-FN characteristics for the Strongly-Strict-MPS-Superstrict family from Monte Carlo simulations for various values of n , m and σ

and the probability that an alternative is suboptimal reduces. As a result, the probability that an alternative is a false positive (which is related to the probability that the alternative is suboptimal) reduces as well. By a similar reason, the probability that an alternative is a false negative increases. As a consequence we see a resultant increase in the expected number of false negatives and reduction in the expected number of false positives. In the extreme case when the number of criteria is increased to infinity, we would expect all alternatives to be optimal and as a result the number of false positives will be zero for any survivor set while the number of false negatives will be large for any set which excludes alternatives at all.

In-between Sets

Figures 16 and 17 show the FP-FN as well as the TP-TN characteristics for the case where the ordering based on in-between sets (as described in Section 6.7.1) is used to produce the survivor set. Note that although both Figure 15 and Figure 16 show the FP-FN characteristics for the same family of filters, they are each parameterized over different parameter and therefore relate to different user needs. Figure 15 is parameterized over values of ϵ and is therefore a performance measure for situations where the user expresses μ values to indicate the assurances he needs in terms of the pairwise survival probabilities of each survivor. On the other hand, Figure 16 is parameterized in terms of the percent survivors picked from the top of the in-between ordering as a survivor set; therefore this plot is a good performance measure for situations where the DM has pruning needs.

Confidence Intervals

A confidence interval is a range of values that has a specified probability of containing the parameter being estimated. The 95% and 99% confidence intervals which have .95 and .99 probabilities of containing the parameter respectively are most commonly used. For example, if the parameter being estimated were q , the 95% confidence interval might look like $15 \leq q \leq 20$. What this means is that the interval between 15 and 20 has a .95 probability of containing q . In other words, if the procedure

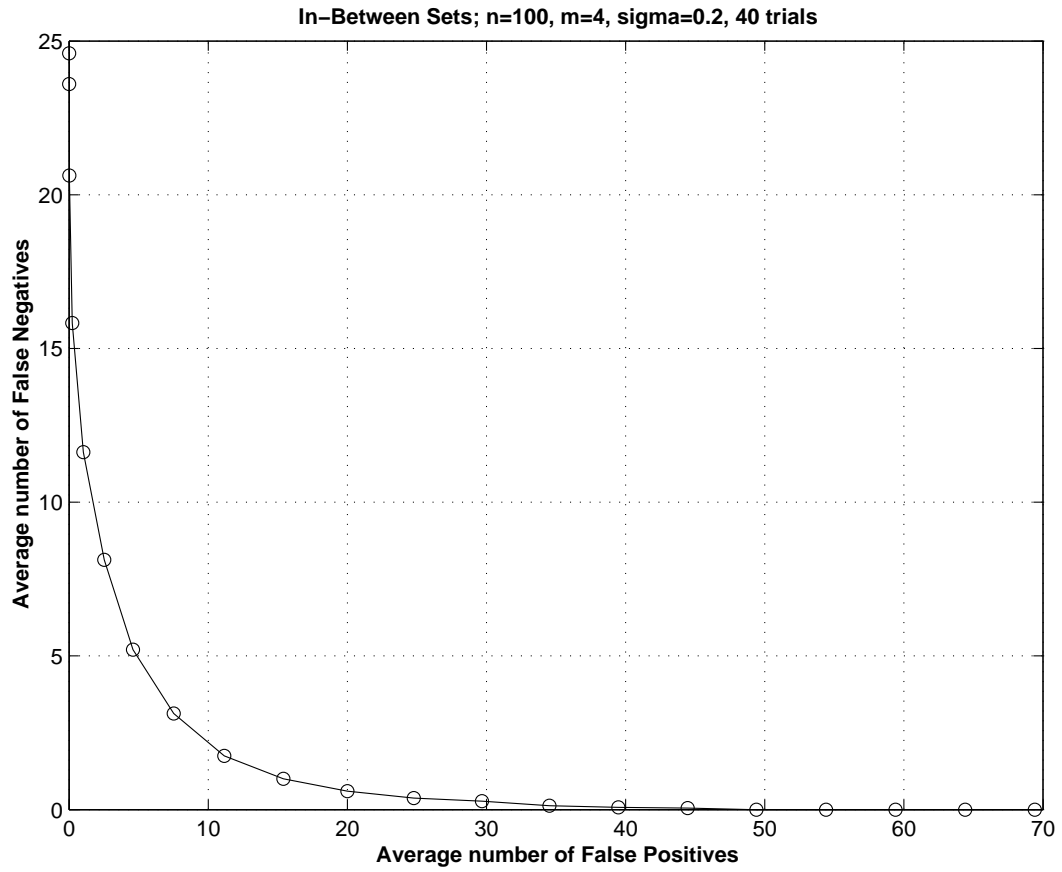


Figure 16: FP-FN characteristics for the in-between sets obtained from various pruning expressions of the DM from Monte Carlo simulations.

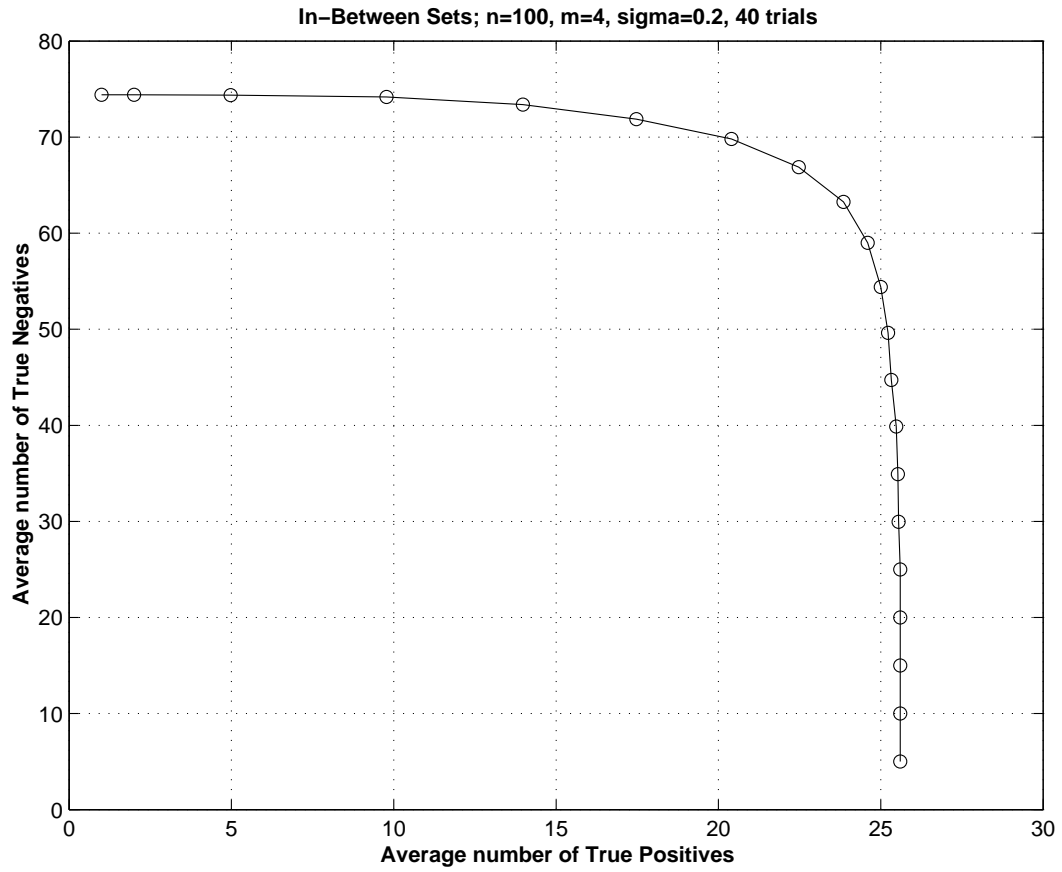


Figure 17: TP-TN characteristics for the in-between sets obtained from various pruning expressions of the DM from Monte Carlo simulations.

for computing a 95% confidence interval is used over and over, 95% of the time the interval will contain the parameter.

Suppose \bar{f}_+ and \bar{f}_- represent one operating point in an FP-FN curve. Then the 95% simultaneous confidence estimates are obtained as:

$$\bar{f}_+ \pm t_{39,0.0125} \frac{\sigma_+}{\sqrt{40}}$$

$$\bar{f}_- \pm t_{39,0.0125} \frac{\sigma_-}{\sqrt{40}}$$

The limits on the mean values indicate that one can be 95% sure that the estimated variable will lie in that range specified by the mean. The variable $t_{39,0.0125}$ is obtained from the Student T-distribution that is often used to estimate the error associated with the mean. The suffix $\{39, 0.0125\}$ indicates the degrees of freedom associated with the estimation (number of trials-1=39) and the figure 0.0125 comes from requiring a simultaneous 95% confidence intervals on two mean components. The quantities σ_+ and σ_- refer to the sample standard deviations of the false positives and false negatives respectively, over the 40 trials conducted.

Figure 18 indicates the simultaneous confidence intervals obtained on the $\{FP, FN\}$ points for one of the cases. These estimates provide additional validity to the statistical robustness of the curves obtained by the simulations. Each box in the figure pertains to one operating point in the curve and the point itself is shown as a dot in the center of the box with the size of the box representing the confidence intervals. The figure also shows that the maximum size of the intervals is less than 2 alternatives both for false positives and false negatives. The confidence intervals are expressed as ranges around the mean values plotted as points in the operating characteristics of the filters.

6.10 Mapping uncertain situations to choice filters

Based on the analysis of the various choice filters, a framework to map choice filters for some decision situations in the presence of uncertainty is shown in Figures 19 and 20, one for situations characterized by bounded noise models and the other for

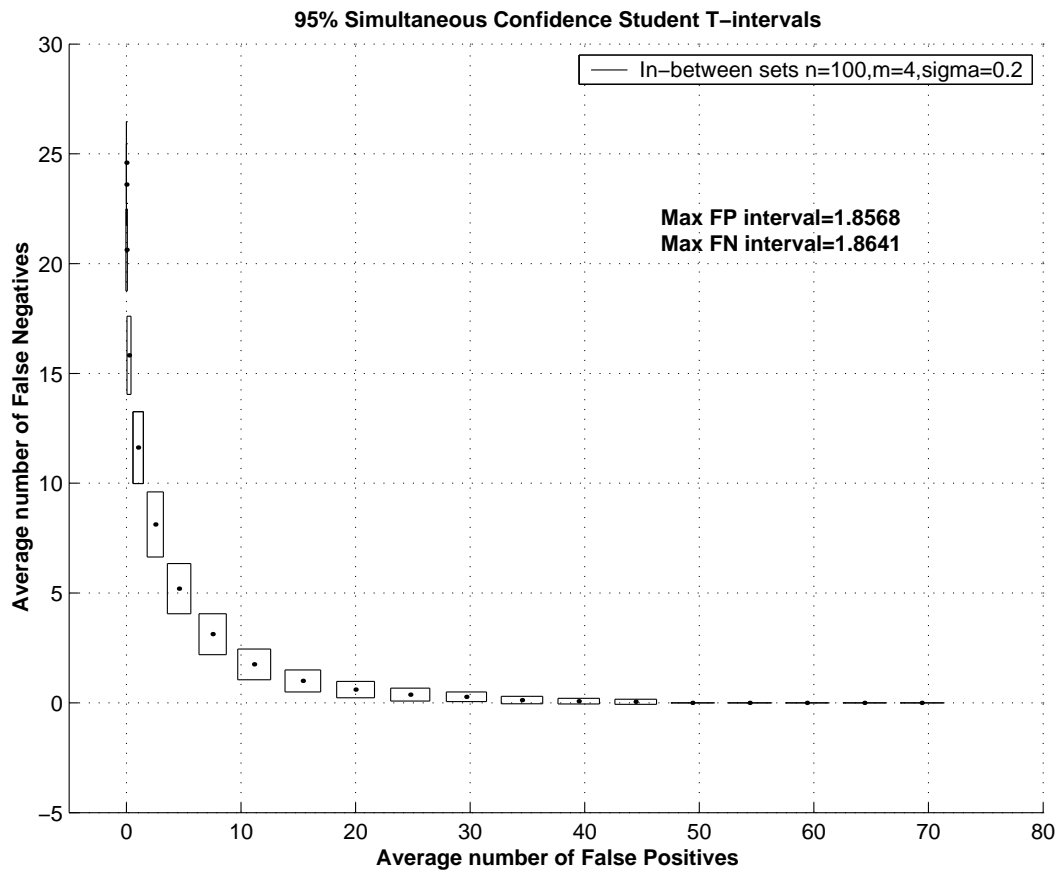


Figure 18: A plot showing Student T-interval 95% simultaneous confidence estimates for each of the points in the FP-FN curve

unbounded noise models. Figure 19 addresses decision situations with bounded noise model, where the bounds are $\pm\epsilon/2$. Using the results from Theorems 6.4 and 6.5, for the two extreme classes of users who require *all* optimal alternatives or those who require *only* optimal alternatives, in the survivor set can be catered to be the application of $SS(\epsilon)$ and $Sts(\epsilon)$ respectively, with the ϵ values being obtained from the bounds indicated for the noise model. For the more general class of users not primarily concerned with optimality so long as bounds can be placed on the FPs and FNs, survivor sets characterized by measured dominance, dd-dominance or ebtp-filters can each be used according to various problem characteristics and other kinds of user-needs.

As discussed earlier, the measured dominance filter produces a survivor set for which the quality of the included false-positives and the excluded false-negatives can both be expressed in terms of the bounds by which they are suboptimal or optimal respectively. A decision-maker who needs fewer alternatives than that produced by the Superstrict filter, or more than that produced by the Strongly-strict filter could therefore use the measured dominance filter.

In case, the decision-maker desires more alternatives than that produced by measured dominance in hope of increasing the number of optimal alternatives in the survivor set, he can then use either dd-dominance or the ebtp-filter, provided the number of criteria are not too large (for large number of criteria, the survivors sets for each of these two filters become identical to the measured dominance survivor set as discussed in Chapter 4). Choice between dd-dominance or the ebtp-filter is based on properties of the noise model and the size of the noise bounds. If the tolerances are small, ebtp filter has a greater chance of producing additional alternatives (over the dominance set) than the dd-dominance filter. If the ϵ values are large, then the choice between the ebtp and dd-dominance filter is based on the DM's expressions of pruning and time-pressure. For large tolerances, the ebtp-filter can produce a lot more alternatives than dd-dominance. Additionally, if the noise model is known to possess central-tendency, then it is recommended that dd-dominance be used instead of ebtp-filter in hope of retaining only those alternatives additional to the measured dominance set, which have a greater chance of being optimal. Figure 20 presents the filter-choice scheme for decision situations characterized by unbounded noise models.

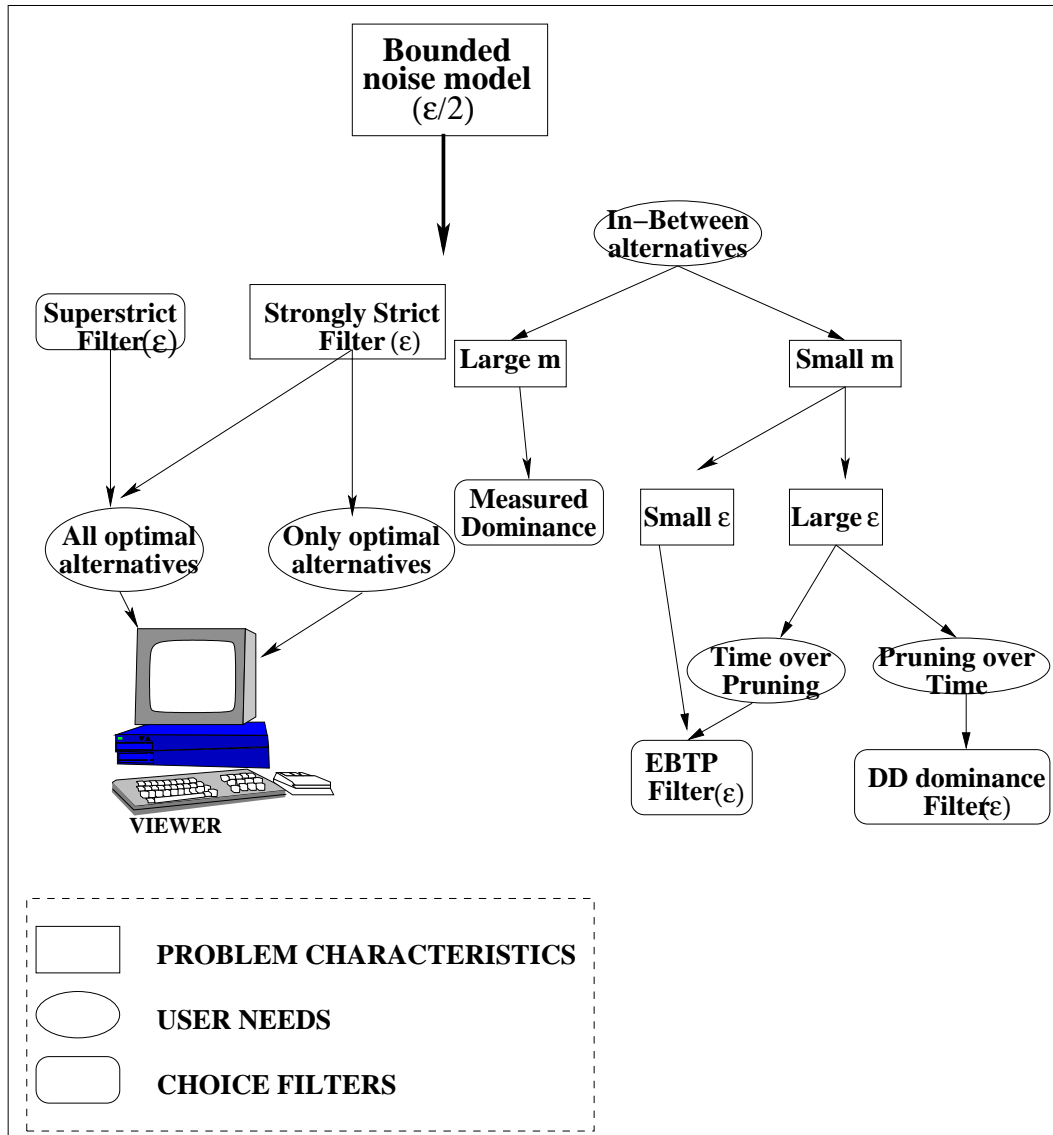


Figure 19: The filter-choice scheme for uncertainty situations with bounded noise models

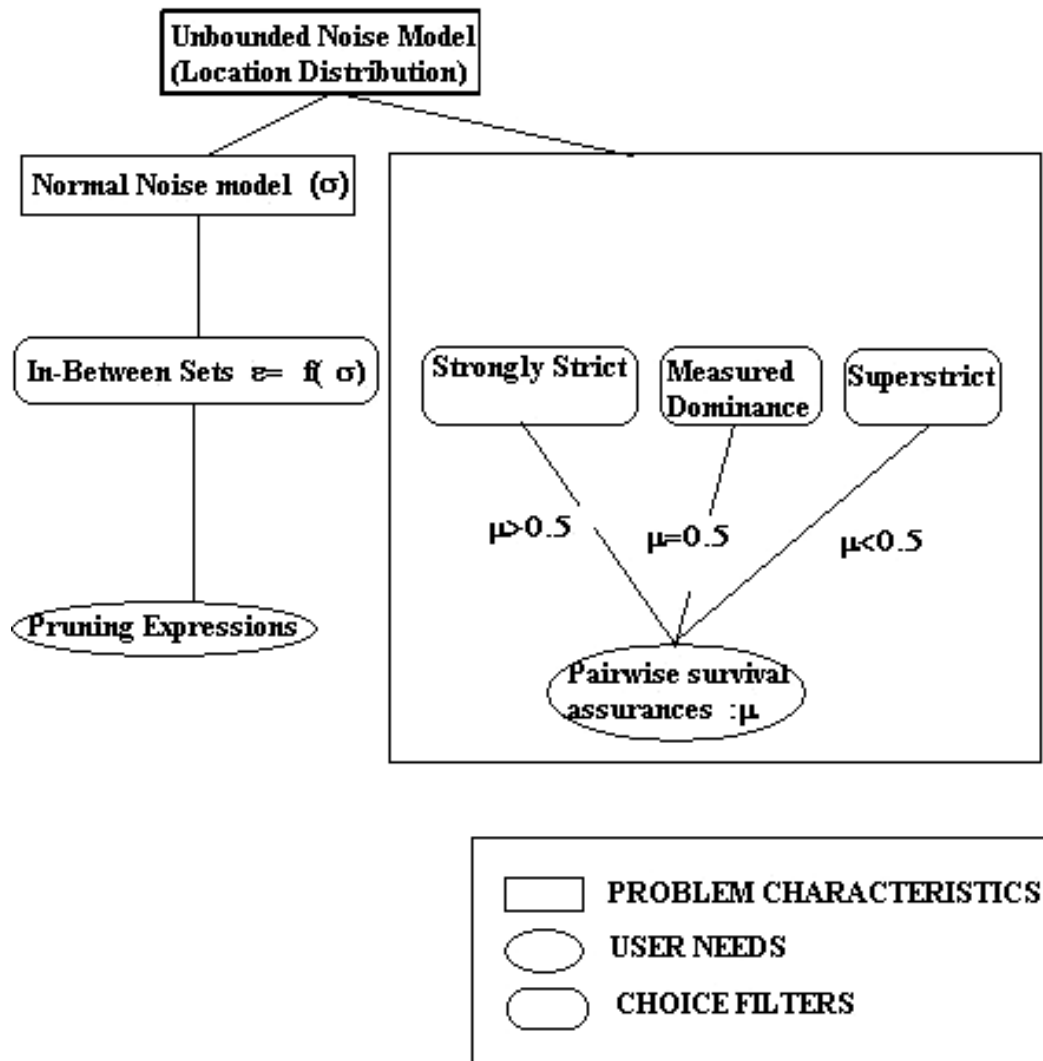


Figure 20: The filter-choice scheme for uncertainty situations with unbounded noise models

If the decision-maker's requirements are better expressed in terms of criteria-wise, or pairwise survival, probabilities of an alternative then the corresponding μ expressions of the decision-maker can be used to infer the corresponding ϵ vector and apply the Strongly-strict, Superstrict, or measured dominance filter based on whether μ is smaller than, equal to, or greater than 0.5. On the other hand if the decision-maker can only provide pruning expressions, and if the noise model is Gaussian, then the ordering based on in-between sets is used to cater to those pruning needs by selecting the corresponding number of alternatives from the top of the ordered set of alternatives.

6.11 Summary

This concludes the problem of mapping choice filters to deal with varying kinds of decision situations in the presence of uncertainty. As is the case with situations without uncertainty, we see that the choice of an appropriate choice filter in a decision situation is again constrained by the kinds of user-needs and the characteristics of the problem and uncertainty. The earlier sections of this chapter indicate that TBDFs are well-suited to catering to problems with bounded noise models. The ability to produce the *necessary set* and the *sufficient set* in addition to being able to produce sets in between the above two kinds of sets imparts the TBDF family with a general applicability to deal with bounded noise uncertainty problems.

We also showed how the ideas of necessity and sufficiency can be extended to uncertainty problems with unbounded noise models described by distributions in the location family. We introduced notions of near-necessity and near-sufficiency to preserve the distinction that the sets produced for the unbounded case do not carry the guarantees that could be given for the bounded case. Nevertheless, we show that different nearly-sufficient and nearly-necessary survivor sets can be produced depending upon how tightly we bound the unbounded noise. The family of Strongly Strict, measured dominance, and Superstrict filters when parameterized over different tolerance values allow us to produce a spectrum of survivor sets, where each survivor set in the spectrum is either a nearly-sufficient, or a nearly-necessary set. Any survivor set in this spectrum can be obtained in correspondence to user expression of pairwise

survival probabilities desired of the survivors. Depending upon whether this quantity is greater than, equal to, or less than 0.5 we showed that it is possible both to select an appropriate filter to use, as well as the tolerance values to be used in the decision rule. This allows us to produce the survivor set of interest to the DM from within the spectrum.

For situations where the DM can only provide his pruning needs, we present an algorithm for ordering the alternatives which allows easy retrieval of any of the survivor sets that occur in the spectrum indicated in Equation 3.2. The DM's pruning needs can now be met by using this ordering to pick out a survivor set of the appropriate size. Moreover, the algorithm described in Section 6.7.2 indicates that this ordering can be produced with a time-complexity equal to that required to produce the dominance set, which is $O(n^2)$ worst-case. The Monte-Carlo simulations allow us indicate the performance of TBDFs in terms of the FP-FN curve.

CHAPTER 7

CONTRIBUTIONS AND FUTURE WORK

The primary goal of the dissertation was to address the issue that requirements posed by the decision-maker and, characteristics of the decision problem typically tend to be present as situation-specific constraints in decision problems. In response, we set out to examine ways in which the S-F-V architecture can be extended to provide situation-specific decision support. In particular, we studied the space of tolerance based dominance filters towards building a framework which can be used to map decision situations to choice filters. The outcome of this study is a mixture of analytical and experimental results that were used to build such a framework for a variety of decision situations. In the next section, we describe the primary contributions of the dissertation.

7.1 Contributions of the Dissertation

The dissertation makes several contributions to the field of multiple criteria decision making and design of decision support systems.

- Analysis based on the survey of literature in behavioral sciences leads to the following contributions:
 - The identification of different kinds of user-preferences based on the manner in which they are best expressed, and on the aspects of the problem that they relate to:

- * *Object preferences* relate to preferences of the decision-maker having to do with the properties of the choice alternatives. Within object preferences are:
 - *Abstract preferences*, which are preferences about the alternatives expressed independent of the alternatives at hand, and
 - *Concrete preferences* which are preferences about the alternatives that are best expressed in the presence of the decision alternatives.
 - * *Process preferences* relate to preferences of the decision-maker that relate to properties of the decision-making process, like speed, nature and size of the solution set.
- The identification of the following desirable properties in order for DSSs to provide effective decision support:
1. DSSs must elicit both object preferences (abstract and concrete preferences) and process preferences, by means which are natural.
 2. DSSs must be responsive to the characteristics of the decision problem at hand.
 3. DSSs must augment the computational capacity of the decision-maker so that the solutions produced respect all of the preferences expressed by the decision-maker.
- An analysis of the applicability of dominance as filter based only upon the measured values (when the actual values are unavailable due to the presence of noise) for the class of decision problems with uncertainty reveals that:
 - The application of measured-dominance produces an equal number of false positives and false negatives.
 - The discovery that one natural extension of the dominance rule, the single pass Strict dominance rule, leads to counterintuitive results. Firstly the number of survivors it produces is not a monotonically increasing function of the magnitude of the tolerances. Secondly, the rule is not transitive, thereby making its solution order-dependent. The ramifications on the use of this filter for decision support

are examined. Also ways to modify it in order to address one or the other problem results in other interesting TBDFs, like the twopass and the multipass strict filters, the dd-dominance filter and the strongly-strict filter.

- An analysis of the ability of TBDFs to cater to various kinds of user-needs in the absence of uncertainty reveals that the different TBDFs can each be used to produce different kinds of *near-optimal* sets of alternatives. These different sets correspond to different notions of *near optimality* that the user might be interested in, when he desires more alternatives in addition to the Pareto set. A framework to map different kinds of *near optimality* to an appropriate TBDF is presented in Figure 8.
- Establishment of the set-inclusion relations among the survivors produced by the different TBDFs. (Corollary 3.2)
- An analysis of how TBDFs can cater to decision situations with uncertainty reveals the following:
 1. For bounded noise models, we show that the Strongly-strict filter produces the *necessary set* (containing *only* optimal alternatives), while the Super-strict filter produces the *sufficient set*(containing *all* optimal alternatives). Each of the other choice filters provide survivor sets between these two extremal sets. The filter-choice scheme for the bounded tolerance case is indicated in Figure 19.
 2. For unbounded, location family noise models, the notions of *necessity* and *sufficiency* become contingent and we introduce the notions of *near-necessity* and *near-sufficiency*. We produce weak bounds on the extent to which these sets are contingent. These bounds can be expressed through quantities that translate the user's probabilistic threshold on the event of the nearly-sufficient set excluding an optimal alternative or on the event of the nearly-necessary set containing a pairwise-dominated alternative. Since the bounds are weak, this technique of producing survivor sets is not operationally useful.

3. As an alternative, we introduce another technique that translates the DM's expression of the probability that an individual survivor is not pairwise dominated by each of the other guys into a corresponding value of μ in the range $[0,1]$.
 4. We show that for the above technique, the appropriate filter to use from the Strongly strict, measured dominance, Superstrict, family is decided by the value of μ itself. We show that a value of $\mu > 0.5$ translates to applying the Strongly strict filter, a value of $\mu = 0.5$ translates to applying the measured dominance filter, and a value of $\mu < 0.5$ translates to applying the Superstrict dominance filter to produce a survivor set.
 5. Equation 3.2 indicates that for the unbounded, normally distributed noise models, it is possible to order the set of alternatives so that all of the in-between sets represented in the equation can be obtained as contiguous sets of alternatives from the top of the ordering. In situations, where the DM can only specify pruning needs, this technique can be used to produce the appropriate subset from the totally ordered set of alternatives.
 6. We present an algorithm which produces this ordering with the same time-complexity as that required to produce the Pareto set. This ordering assumes the noise models to be normally distributed with deviation σ 's specified. Based on this ordering, pruning expressions of the decision-maker can be mapped onto a corresponding in-between set.
- Results based on Monte-Carlo simulations show the FP-FN performance of the ordering based on in-between sets. Confidence estimates are also computed for the individual operating points obtained by the simulations, to strengthen the validity of the statistical indications.

7.2 Future Work

In this section, we discuss some issues related to the dissertation but which require further exploration as research problems.

7.2.1 Decision situations with very large number of criteria

We have already mentioned that the decision problems with large number of criteria are the most troublesome for the S-F-V architecture. In general, a large m problem is an intrinsically difficult decision problem and perhaps the solution lies in the problem formulation rather than in the manner in which support is provided for its solution. The dominance filter can be expected to produce some pruning, while interaction with the Viewer might require the decision-maker to focus on a lot of information in order to ensure that he is taking the consequences of his graphical selections into account before deciding to proceed with the next step.

Traditionally, processing based on a subset of the information like lexicographic ordering, elimination by aspects, or majority of confirming dimensions are used. But these techniques can be shown to produce solutions which need not be the best from the viewpoint of the decision-maker. However, we also know from the results discussed in Chapter 2 related to correlations in criteria, that most of the criteria in a large set of criteria are expected to be positively correlated. This automatically suggests that all strongly and positively correlated criteria could be aggregated into a single criteria. For example, if there are a number of criteria which are different kinds of efficiency measures, then these could be aggregated (by the use of a suitable aggregation function, like linear weighting) to produce a single efficiency criterion.

It can be shown that as long as the aggregating function is a monotonically increasing function of its components and if an alternative A dominates another alternative B in the aggregated criteria, then A cannot be dominated by B under the disaggregated criteria set. In other words, this technique can only produces false negatives with respect to the optimal set based on the original set of criteria. The question is whether this false negative rate can be reduced by using something like the Super-strict filter (or even dd-dominance or ebt_p) on the aggregated criteria rather than the dominance filter. It would be interesting to analyze what kinds of user concerns in terms of the disaggregated criteria map to the tolerances applied to aggregated criteria.

7.2.2 Representing uncertainty in the Viewer

A large part of the dissertation dealt with analyzing various kinds of choice filters which can deal with uncertainty in the criteria values to produce survivor sets for which assurances of various sorts can be given. However, in the context of the S-F-V architecture, the final stage of the decision-making process might involve the use of the Viewer, where the user is allowed to plot the Filter survivors as Cartesian points using their measured criteria values. Since most of the choice made by the decision-maker while using the Viewer will be based on the relative location of the individual points in the plots, representation of the alternatives as points based on the inaccurate criteria values might mislead the decision-maker. Therefore it is desirable that the uncertainty associated with the criteria values be somehow represented in the Viewer as well.

One possible alternative is to represent points as boxes where the size of the boxes could represent the actual range of values for the alternative, if the noise is bounded. However, this can lead to problems like occlusion and clutter in the plot so that the user might find it confusing to interact with the visible points. It is an interesting question as to what kinds of visual primitives can be used to achieve this efficiently. It will also entail modifying the selection and cross-linking mechanisms of the Viewer.

7.2.3 The Filter as an adviser/critic

The Filter module in the S-F-V architecture can act as an *adviser* or as a *critic* for choices made by a decision-maker. While the Seeker-Filter-Viewer architecture can be used as an aid throughout the decision-making process starting from generation of the decision alternatives, one can imagine decision situations where the decision-maker has already made the decision (with or without using any additional support) and is now concerned with the relative *quality* of his final choice(s) compared to the overall set of alternatives.

For such a decision situation, the Filter can act as an adviser by producing competing alternatives from the original set which might be of interest to the user. For example, if the final choice(s) of the DM happen to be suboptimal, the Filter can produce the dominators of the final choice(s) as competing alternatives. This will

help in minimizing the irrationality of the decision-making process employed by the DM. The relevant question is what sorts of competing alternatives can be expected to catch the DM's attention and how TBDFs can be used to produce these various kinds of *competing* alternatives.

APPENDIX A

PROOFS OF THEOREMS AND COROLLARIES

A.1 Introduction

This appendix contains proofs for all the Theorems, and Corollaries stated in the text of the dissertation. The order in which they appear below is the same as the order in which they appear in the text. Also, they are numbered according to the numbering in the text, by the chapter numbers. For each proof, we first state the claim being proved at the beginning and then present the proof. The terminology used in the proofs is identical to the one used and described in Section 3.2.1.

A.2 Proofs for Chapter 2

Theorem 2.1

Every distinct¹⁶ survivor of an m -criteria dominance is also a survivor of $(m+k)$ -criteria dominance, $k \geq 0$, where the $(m+k)$ -criteria dominance uses all the criteria used by the m -criteria dominance.

Proof:

For all distinct dominance survivors we can write,

$$C_j \in S(D) \iff \forall C_k (\exists r (x_{jr} > x_{kr}))$$

¹⁶By a distinct alternative is meant one for which no other alternative is coincident with it on all criteria

Suppose $C_i \in S(D)$ where D uses m criteria. Since C_i survives D , the above relation holds for C_i and we can write,

$$\forall C_k (\exists r (x_{ir} > x_{kr}))$$

With the introduction of any number of additional criteria, the above relation will continue to hold. Hence C_i will survive D even if it uses $m + k$ criteria chosen by the introduction of k criteria to the previous set of m criteria, $k \geq 0$.

A.3 Proofs for Chapter 3

Theorem 3.1

The Strict Dominance rule cannot be strongly intransitive for a decision problem with $m < 3$.

Proof:

Consider three alternatives C_i , C_j , and C_k , each evaluated in terms of two criteria, say c_1 and c_2 .

Suppose also that these three alternatives form a strongly intransitive loop under the strict dominance rule applied considering the two criteria above. Let this loop be such that C_i strictly dominates C_j strictly dominates C_k strictly dominates C_i . We show that this assumption is contradictory under all possible scenarios.

Since C_i strictly dominates C_j , therefore C_i must be ϵ -better than C_j on one of the two criteria being considered. Without loss of generality let this criterion be c_1 . Therefore we have,

$$x_{i1} > x_{j1} + \epsilon. \tag{A.1}$$

Also, since C_j strictly dominates C_k , there must be a criterion in which C_j is ϵ -better than C_k . This criterion could be either c_1 or c_2 . Let us consider this one case at a time:

1. Suppose C_j is ϵ -better than C_k on criterion c_1 .

Then we must have,

$$x_{j1} > x_{k1} + \epsilon. \tag{A.2}$$

From Equations A.1 and A.2 we must have,

$$x_{i1} > x_{k1} + \epsilon. \quad (\text{A.3})$$

But this contradicts the assumption that C_k *strictly dominates* C_i because if C_i is ϵ -better than C_k on c1, then C_k cannot *strictly dominate* C_i (considering the second clause of the strict dominance decision rule). Incidentally, this means that if the only criterion being considered was c1, then we cannot have strong intransitivity. In other words, this proves the claim for $m = 1$.

2. Conversely suppose C_j is ϵ -better than C_k on criterion c2.

Then we must have,

$$x_{j2} > x_{k2} + \epsilon. \quad (\text{A.4})$$

In other words, we have Equations A.1 and A.4 holding to be true. Now we have assumed that C_k *strictly dominates* C_i . This means that C_k must be ϵ -better than C_i on at least one of the criterion. Here again, we have to consider two cases:

- (a) Suppose that C_k is ϵ -better than C_i on criterion c1.

Then we must have,

$$x_{k1} > x_{i1} + \epsilon. \quad (\text{A.5})$$

From Equations A.1 and A.5 we further have,

$$x_{k1} > x_{j1} + \epsilon. \quad (\text{A.6})$$

However, this contradicts our original assumption that C_j *strictly dominates* C_k considering the strict dominance rule.

- (b) Conversely suppose that C_k is ϵ -better than C_i on criterion c2.

Then we must have

$$x_{k2} > x_{i2} + \epsilon. \quad (\text{A.7})$$

In other words, Equations A.1, A.4, and A.7 hold true. But they imply the following,

$$x_{j2} > x_{i2} + \epsilon. \quad (\text{A.8})$$

This again contradicts our assumption that C_i strictly dominates C_j .

Hence, we reach a contradiction upon considering all possible scenarios based on our initial assumption that the three alternatives form a strongly intransitive loop. Therefore, our initial assumption that the three alternatives form an intransitive loop based on 2 criteria must be incorrect. Since the claim for $m = 1$ was also proved above we have that for $m < 2$, we cannot have strong intransitivity by applying the strict dominance rule.

This completes the proof.

Theorem 3.2

For $\tilde{\mathcal{X}} = \{C, \tilde{X}\}$,

$$\emptyset = S(S_{ts}(\epsilon_k), \tilde{\mathcal{X}}) \subseteq \cdots S(S_{ts}(\epsilon_3), \tilde{\mathcal{X}}) \subseteq S(S_{ts}(\epsilon_2), \tilde{\mathcal{X}}) \subseteq S(S_{ts}(\epsilon_1), \tilde{\mathcal{X}}) = S(D, \tilde{\mathcal{X}}).$$

where,

$$\epsilon_k > \cdots > \epsilon_2 > \epsilon_1 \geq 0.$$

Proof:

By setting ϵ_1 to 0, it is clear that the first pass of Strongly Strict filter, which uses the Strict dominance rule, produces the measured dominance survivors. Since the second pass eliminates each such first-pass survivor which has some other alternative ϵ -indistinguishable on all criteria with respect to it, the second pass when applied on the measured dominance set will not eliminate any additional alternatives *provided* no two alternatives have identical measured values in all criteria. Hence by construction, we see that,

$$\exists \epsilon_1 S(S_{ts}(\epsilon_1)) = S(D, \tilde{\mathcal{X}}). \quad (\text{A.9})$$

Suppose we set each of the m components of the ϵ_k vector as follows,

$$(\text{for } i = 1 \text{ to } m) \epsilon_k[i] = \max_{j=1:n} [x_{ji}] - \min_{j=1:n} [x_{ji}].$$

Then, no single alternative will survive the Strongly Strict filter because for the above setting of tolerances, every alternative will be ϵ -indistinguishable on all criteria with respect to every other alternative in the set and as a result no single alternative will survive the second pass. Therefore we have,

$$\exists \epsilon_k S(Sts(\epsilon_k)) = \emptyset. \quad (\text{A.10})$$

Finally, suppose that ϵ_1 and ϵ_2 are two tolerance vectors such that $\epsilon_1 < \epsilon_2$, meaning that each component of the vector ϵ_1 is lesser than or equal to the corresponding components in the vector ϵ_2 with at least one strict inequality. Now,

Suppose $C_i \in S(Sts(\epsilon_2))$. This implies that C_i is ϵ_2 -distinguishable with respect to every other alternative on some criterion or the other. This further implies that C_i is ϵ_1 -distinguishable with respect to every other alternative on some criterion or the other, since $\epsilon_1 < \epsilon_2$. Hence $C_i \in S(Sts(\epsilon_1))$. Thus we have,

$$C_i \in S(Sts(\epsilon_2)) \Rightarrow C_i \in S(Sts(\epsilon_1)).$$

Or in other words,

$$S(Sts(\epsilon_2)) \subseteq S(Sts(\epsilon_1)). \quad (\text{A.11})$$

where $\epsilon_1 < \epsilon_2$. Since our choice of ϵ_1, ϵ_2 was arbitrary, we can use Equations A.9, A.10, and A.11 and write,

$$\emptyset = S(Sts(\epsilon_k), \tilde{\mathcal{X}}) \subseteq \cdots S(Sts(\epsilon_3), \tilde{\mathcal{X}}) \subseteq S(Sts(\epsilon_2), \tilde{\mathcal{X}}) \subseteq S(Sts(\epsilon_1), \tilde{\mathcal{X}}) = S(D, \tilde{\mathcal{X}}).$$

where,

$$\epsilon_k > \cdots > \epsilon_2 > \epsilon_1 \geq 0.$$

This completes the proof.

Theorem 3.3

For $\tilde{\mathcal{X}} = \{C, \tilde{X}\}$,

$$\begin{aligned} S(D, \tilde{\mathcal{X}}) &= S(SS(\epsilon_1), \tilde{\mathcal{X}}) \subseteq S(SS(\epsilon_2), \tilde{\mathcal{X}}) \subseteq S(SS(\epsilon_3), \tilde{\mathcal{X}}) \cdots \\ &\cdots \subseteq S(SS(\epsilon_k), \tilde{\mathcal{X}}) = C. \end{aligned}$$

where,

$$0 \leq \epsilon_1 < \epsilon_2 \cdots < \epsilon_k$$

Proof:

According to the Superstrict dominance rule, we say

$$(C_i \text{ superstrictly dominates } C_j) \iff \exists k(x_{ik} > x_{jk} + \epsilon) \wedge \forall k(x_{ik} \geq x_{jk} + \epsilon).$$

If we set all the m components of the ϵ vector to zeroes we get the dominance rule. Hence we have,

$$S(D, \tilde{\mathcal{X}}) = S(SS(\epsilon_1), 0 \leq \epsilon_1). \quad (\text{A.12})$$

Suppose we set one of the m components of the ϵ_k vector, say the i th component, as follows,

$$\epsilon_k[i] = \max_{j=1:n} [x_{ji}] - \min_{j=1:n} [x_{ji}].$$

Then, no alternative in $\tilde{\mathcal{X}}$ will be able to superstrictly-dominate another alternative since the strict inequality clause in the superstrict rule will not be satisfied along the i th criterion. Hence *all* n alternatives will survive the application of Superstrict dominance with the above value of tolerances. In other words,

$$S(SS(\epsilon_k, \tilde{\mathcal{X}})) = C. \quad (\text{A.13})$$

Finally, suppose that ϵ_1 and ϵ_2 are two tolerance vectors such that $\epsilon_2 > \epsilon_1$, meaning that each component of the vector ϵ_2 is greater than or equal to the corresponding components in the vector ϵ_1 . Now,

$$\begin{aligned} C_i &\in S(SS(\epsilon_1, \tilde{\mathcal{X}})) \Rightarrow \neg \exists C_j \forall k (x_{jk} > x_{ik} + \epsilon_1). \\ &\Rightarrow \neg \exists C_j \forall k (x_{jk} > x_{ik} + \epsilon_2). \\ &\Rightarrow C_i \in S(SS(\epsilon_2, \tilde{\mathcal{X}})). \end{aligned} \quad (\text{A.14})$$

In other words,,

$$(C_i \in S(SS(\epsilon_1, \tilde{\mathcal{X}})) \rightarrow (C_i \in S(SS(\epsilon_2, \tilde{\mathcal{X}}))).$$

Since our choice of the vectors, ϵ_1 and ϵ_2 was arbitrary, this proves that,

$$S(SS(\epsilon_1), \tilde{\mathcal{X}}) \subseteq S(SS(\epsilon_2)). \quad (\text{A.15})$$

for $\epsilon_1 < \epsilon_2$. We can now, generalize the results expressed in Equations A.12, A.13 and A.15 to get the desired result:

$$S(D, \tilde{\mathcal{X}}) = S(SS(\epsilon_1), \tilde{\mathcal{X}}) \subseteq S(SS(\epsilon_2), \tilde{\mathcal{X}}) \subseteq S(SS(\epsilon_3), \tilde{\mathcal{X}}) \cdots \subseteq S(SS(\epsilon_k), \tilde{\mathcal{X}}) = C.$$

where $0 \leq \epsilon_1 < \epsilon_2 < \cdots < \epsilon_k$.

This completes the proof.

Theorem 3.4

Every survivor of the dominance filter is a survivor of dd-dominance filter. i.e.,

$$C_i \in S(D, \tilde{\mathcal{X}}) \Rightarrow C_i \in S(DD(\epsilon), \tilde{\mathcal{X}})$$

Proof:

Suppose that

$$C_i \notin S(DD(\epsilon), \tilde{\mathcal{X}}). \quad (\text{A.16})$$

This means that there is some alternative C_j which dd-dominates C_i i.e. a C_j such that,

$$\forall k \tilde{x}_{jk} \geq \tilde{x}_{ik} \wedge \exists p \tilde{x}_{jp} > \tilde{x}_{ip} + \epsilon.$$

This further implies that

$$\forall k \tilde{x}_{jk} \geq \tilde{x}_{ik} \wedge \exists p \tilde{x}_{jp} > \tilde{x}_{ip}.$$

Which implies that the alternative C_j dominates C_i . Or that,

$$C_i \notin S(D, \tilde{\mathcal{X}}) \quad (\text{A.17})$$

Applying contraposition on Equations A.16 and A.17 the theorem follows.

This completes the proof.

Theorem 3.5

Every survivor of Twopass Strict Dominance is a survivor of dd dominance, for the same tolerance value i.e.,

$$C_i \in S(St2(\epsilon)) \Rightarrow C_i \in S(DD(\epsilon)).$$

Proof:

Suppose there is some alternative C_i which survives Twopass Strict dominance but does not survive dd-dominance.

Since C_i does not survive dd-dominance there must be some alternative C_j which dd-dominates it. In other words,

$$\forall a \tilde{x}_{ja} \geq \tilde{x}_{ia} \wedge \exists b \tilde{x}_{jb} > \tilde{x}_{ib} + \epsilon.$$

Using the Strict dominance rule we see that the above condition implies that C_j *strictly-dominates* C_i . Since Twopass Strict is the set of Strictly undominated alternatives, C_i cannot be a survivor of Twopass Strict dominance. Hence our initial assumption must be wrong.

This completes the proof.

Theorem 3.6

Every survivor of the ϵ -box Twopass(ebtp) filter is a survivor of the Superstrict dominance filter for the same tolerance value i.e.,

$$C_i \in S(EB(\epsilon)) \Rightarrow C_i \in S(SS(\epsilon)).$$

Proof:

An alternative C_i can be survivor of the ebtp-filter by one of two possible ways. Either it is a dominance survivor so that it survives the first pass, or it is within the

ϵ -box with respect to some measured dominance survivor. We consider each of these types of survivors separately and show that the claim hold for both kinds of survivors.

Case 1: Suppose C_i survives the ebtp-filter because it is a survivor of dominance. Assume, for contradiction that C_i does not survive Superstrict dominance. This means that there must be some other alternative C_j which superstrictly dominates C_i . That is,

$$\forall p \tilde{x}_{jp} \geq \tilde{x}_{ip} + \epsilon \wedge \exists q \tilde{x}_{jq} > \tilde{x}_{iq} + \epsilon.$$

This implies that

$$\forall p \tilde{x}_{jp} \geq \tilde{x}_{ip} \wedge \exists q \tilde{x}_{jq} > \tilde{x}_{iq}.$$

This means that the alternative C_j dominates C_i . However, this contradicts our assumption that C_i is a dominance survivor. Hence our assumption that C_i will not survive Superstrict dominance must be wrong.

Case 2: Suppose that C_i survives the ebtp-filter not by its first pass but by its second pass. This means that C_i must within the ϵ -box form some Pareto alternative. Let C_j be such an alternative. Note that since C_j is a dominance survivor, it must also be a survivor of the ebtp-filter.

$$\forall k \tilde{x}_{jk} < \tilde{x}_{ik} + \epsilon. \tag{A.18}$$

Suppose we assume, for contradiction, that C_i does not survive Superstrict dominance. Then we know that there must be some alternative C_t which superstrictly dominates C_i . In other words,

$$\forall p \tilde{x}_{tp} \geq \tilde{x}_{ip} + \epsilon \wedge \exists q \tilde{x}_{tq} > \tilde{x}_{iq} + \epsilon. \tag{A.19}$$

From Equations A.18 and A.19 we can infer,

$$\forall p \tilde{x}_{tp} \geq \tilde{x}_{jp} \wedge \exists q \tilde{x}_{tq} > \tilde{x}_{jq}.$$

This implies that the alternative C_t dominates C_j . This is contrary to our assumption that C_j is a survivor of dominance. Hence our initial assumption, that C_i does not survive superstrict dominance must be incorrect. So, the theorem follows for the second case as well.

Therefore, for all survivors of the ebtp-filter we must have that they survive superstrict dominance as well.

This completes the proof.

Theorem 3.7

A survivor of the Superstrict dominance filter need not be a survivor of the ebtp-filter for the same tolerance value.

Proof:

All that is needed is a counterexample. Consider the 2 alternative, 2 criteria decision problem shown below,

	x	y
C_1	4	4
C_2	3	0

Suppose we use the tolerance vector $\{1.1, 1.1\}$. Clearly, only C_1 will survive the ebtp-filter. However both C_1 and C_2 will survive superstrict dominance.

This completes the proof.

Theorem 3.8

Every survivor of dd-dominance filter is a survivor of the ebtp-filter for the same tolerance value i.e.,

$$C_i \in S(DD(\epsilon)) \Rightarrow C_i \in S(EB(\epsilon)).$$

Proof:

Suppose, $C_i \in S(DD(\epsilon))$.

Since dd-dominance requires dominance and an additional ϵ -distinguishability on some criterion, we must have either that C_i is not dominated by any alternative (so that it is a dominance survivor) or that C_i is ϵ -indistinguishable on all criteria with respect to its dominators. Either way, we get that C_i must survive the ebtp-filter.

This completes the proof.

Theorem 3.9

A survivor of the ebt_p-filter need not be a survivor of the dd-dominance filter for the same tolerance value.

Proof:

All that is needed is a counterexample. Consider the 3 alternative, 2 criteria decision problem shown below,

	x	y
C_1	5	4
C_2	4	5
C_3	3	4

Suppose we use the tolerance vector $\epsilon = \{1.1, 1.1\}$. We see that,

$$S(EB(\epsilon)) = \{C_1, C_2, C_3\} \text{ while,}$$

$$S(DD(\epsilon)) = \{C_1, C_2\}.$$

This completes the proof.

Corollary 3.1

Every survivor of the Strongly Strict filter is a dominance survivor, i.e.,

$$S(Sts(\epsilon), \tilde{\mathcal{X}}) \in S(D, \tilde{\mathcal{X}}).$$

Proof:

Rewriting Equation A.35 we have,

$$(\forall C_j \in S(Sts(\epsilon)) (\forall C_i \in C) : \exists t(\tilde{x}_{jt} > \tilde{x}_{it} + \epsilon).$$

which implies that

$$(\forall C_j \in S(Sts(\epsilon)) (\forall C_i \in C) : \exists t(\tilde{x}_{jt} > \tilde{x}_{it}).$$

This implies that every survivor of the Strongly Strict filter has to be a dominance survivor.

This completes the proof.

A.4 Proofs for Chapter 6

Theorem 6.1

If the actual criteria values and the measured criteria values belong to continuous distributions then the application of measured dominance produces a survivor set for which the following is true:

$$E(|f_+(D, \tilde{\mathcal{X}})|) = E(|f_-(D, \tilde{\mathcal{X}})|). \quad (\text{A.20})$$

Proof:

In their paper [12], Calpine and Golding derive an expression for the expected number of dominance survivors as a function of the number of alternatives n and number of criteria m . They further show the general applicability of this expression to all $n \times m$ set of alternatives, so long as the criteria are independent and the values come from continuous distributions.

More specifically, this means that the application of the dominance filter to the set X of actual criteria values and the set \tilde{X} of measured criteria values, can be both expected to produce the same number of survivors. i.e.,

$$E(|S(D, \mathcal{X})|) = E(|S(D, \tilde{\mathcal{X}})|). \quad (\text{A.21})$$

By the definitions of false positives and false negatives we can write,

$$|S(D, \mathcal{X})| = |S(D, \tilde{\mathcal{X}})| + f_+(D, \tilde{\mathcal{X}}) - f_-(D, \tilde{\mathcal{X}}). \quad (\text{A.22})$$

Now, for random variables X , Y and Z we know that,

$$(X = Y + Z) \rightarrow E(X) = E(Y) + E(Z).$$

Hence, for Equation A.22 we can write,

$$E(|S(D, \mathcal{X})|) = E(|S(D, \tilde{\mathcal{X}})|) + E(f_+(D, \tilde{\mathcal{X}})) - E(f_-(D, \tilde{\mathcal{X}})). \quad (\text{A.23})$$

From Equation A.21 and Equation A.23 we get,

$$E(|f_+(D, \tilde{\mathcal{X}})|) = E(|f_-(D, \tilde{\mathcal{X}})|). \quad (\text{A.24})$$

which is the desired result.

Theorem 6.2

For location family noise models, if alternative C_i dominates alternative C_j in measured dominance then the probability that C_i is not actually dominated by C_j is greater than the probability that C_j is not actually dominated by C_i . In other words,

$$[C_i \text{ measured dominates } C_j] \rightarrow [P(E_i, C_j) > P(E_j, C_i)].$$

Proof:

From Equation 6.27 we have,

$$P(E_i, C_j) = 1 - \prod_{p=1}^{p=m} P(x_{jp} > x_{ip}).$$

This can be written as,

$$P(E_i, C_j) = 1 - \epsilon_{i1}\epsilon_{i2} \cdots \epsilon_{im}.$$

Similarly,

$$P(E_j, C_i) = 1 - \prod_{p=1}^{p=m} P(x_{ip} > x_{jp})$$

or,

$$P(E_j, C_i) = 1 - \epsilon_{j1}\epsilon_{j2} \cdots \epsilon_{jm}.$$

Now, since C_i dominates C_j in measured dominance, we must have,

$$\forall p \tilde{x}_{ip} \geq \tilde{x}_{jp} \wedge \exists q \tilde{x}_{iq} > \tilde{x}_{jq}.$$

From Equation 6.8 we get therefore that,

$$\forall p P(x_{ip} > x_{jp}) \geq 0.5 \wedge \exists q P(x_{iq} > x_{jq}) > 0.5.$$

Or,

$$\forall p P(x_{jp} > x_{ip}) \leq 0.5 \wedge \exists q P(x_{jq} > x_{iq}) < 0.5.$$

This allows us to conclude that,

$$\forall r \epsilon_{jr} \geq \epsilon_{ir} \wedge \exists s \epsilon_{js} > \epsilon_{is}.$$

Or,

$$\prod_{r=1}^{r=m} \epsilon_{jr} > \prod_{r=1}^{r=m} \epsilon_{ir}.$$

That is,

$$[1 - \prod_{r=1}^{r=m} \epsilon_{jr}] < [1 - \prod_{r=1}^{r=m} \epsilon_{ir}].$$

which implies that,

$$P(E_j, C_i) < P(E_i, C_j).$$

Therefore we can conclude,

$$[C_i \text{ measured dominates } C_j] \rightarrow [P(E_i, C_j) > P(E_j, C_i)].$$

This completes the proof.

Theorem 6.3

For location family noise models, if alternative C_i is a survivor of measured dominance then

$$\forall j P(E_i, C_j) > 0.5.$$

where $P(E_i, C_j)$ is the probability that the alternative C_i is not actually dominated by alternative C_j

Proof:

We are given C_i which is a survivor of measured dominance. Let us consider

another alternative, C_j , from the remaining $n - 1$ alternatives. We know that the probability that C_i is not dominated by alternative C_j in terms of the actual values is given by,

$$P(E_i, C_j) = 1 - \prod_{p=1}^{p=m} P(x_{jp} > x_{ip}). \quad (\text{A.25})$$

Since C_i is a survivor of measured dominance exhaustively either,

$$[C_j \text{ is not dominated by } C_i]$$

or,

$$[C_i \text{ dominates } C_j].$$

Case 1: C_i is not dominated by C_j This implies,

$$\exists p (\tilde{x}_{ip} > \tilde{x}_{jp}). \quad (\text{A.26})$$

$$\exists q (\tilde{x}_{jq} > \tilde{x}_{iq}). \quad (\text{A.27})$$

where \tilde{x}_{rs} is the measured value of alternative C_r on criterion s .

From Equation 6.8 we have therefore,

$$\exists q P(x_{jq} > x_{iq}) > 1/2. \quad (\text{A.28})$$

$$\exists p P(x_{ip} > x_{jp}) > 1/2 \text{ which implies that,} \quad (\text{A.29})$$

$$\exists p P(x_{jp} > x_{ip}) < 1/2. \quad (\text{A.30})$$

From the last equation we can infer that,

$$\prod_{p=1}^{p=m} P(x_{jp} > x_{ip}) < 1/2.$$

This implies that,

$$1 - \prod_{p=1}^{p=m} P(x_{jp} > x_{ip}) > 1/2.$$

This, from Equation A.25 implies that

$$P(E_i, C_j) > 1/2.$$

Case 2: C_i dominates C_j This implies,

$$\forall p (\tilde{x}_{ip} \geq \tilde{x}_{jp}) \wedge \exists q (\tilde{x}_{iq} > \tilde{x}_{jq}). \quad (\text{A.31})$$

From Equation 6.8 we have therefore,

$$\forall p P(x_{ip} > x_{jp}) \geq 1/2 \wedge \exists q P(x_{iq} > x_{jq}) > 1/2. \quad (\text{A.32})$$

This implies that

$$\forall p P(x_{jp} > x_{ip}) \leq 1/2 \wedge \exists q P(x_{jq} > x_{iq}) < 1/2.$$

Or,

$$1 - \prod_{p=1}^{p=m} P(x_{jp} > x_{ip}) > 1/2.$$

which from Equation A.25 implies that

$$P(E_i, C_j) > 1/2.$$

Thus the result is proved for all possible choices of a second alternative C_j . Therefore, we must have, in general for all measured dominance survivors that,

$$P(E_i, C_j) > 1/2.$$

This completes the proof.

Theorem 6.4

For bounded noise model $M(\epsilon/2)$, the application of Superstrict dominance filter $SS(\epsilon)$ produces the sufficient set.

Proof:

Let N be the set of alternatives which do not survive the application of $SS(\epsilon)$, where ϵ is the tolerance vector obtained from the bounded noise model. This implies that,

$$\forall(C_i \in N) \exists(C_j \in S(SS(\epsilon))) : [C_j \text{ superstrictly dominates } C_i].$$

$$\forall(C_i \in N) \exists(C_j \in S(SS(\epsilon))) : \exists r(\tilde{x}_{jr} > \tilde{x}_{ir} + \epsilon) \wedge \forall s(\tilde{x}_{jr} \geq \tilde{x}_{ir} + \epsilon). \quad (\text{A.33})$$

For the bounded noise model, we know from Equation 6.4 that

$$\tilde{x} > \tilde{y} + \epsilon \rightarrow x > y.$$

Therefore we can rewrite Equation A.33 as,

$$\forall(C_i \in N) \exists(C_j \in S(SS(\epsilon))) : \exists r(x_{jr} > x_{ir}) \wedge \forall s(x_{jr} \geq x_{ir}). \quad (\text{A.34})$$

This implies that every non-survivor of the Superstrict dominance filter is such that some survivor of the filter actually-dominates it. Therefore, none of the non-survivors of the Superstrict filter can be optimal. In other words, for the bounded noise model, the application of the Superstrict filter produces no false negatives.

This completes the proof.

Theorem 6.5

For bounded noise uncertainty model $M(\epsilon/2)$, the Strongly Strict filter $Sts(\epsilon)$ produces the necessary set.

Proof:

From its decision rule we know that every survivor of the Strongly Strict filter is such that it is ϵ -distinguishable with respect to every other alternative on some criterion or the other i.e,

$$(\forall C_j \in S(Sts(\epsilon))) (\forall C_i \in C) : \exists t(\tilde{x}_{jt} > \tilde{x}_{it} + \epsilon). \quad (\text{A.35})$$

From Equation 6.4, this implies that,

$$(\forall C_j \in S(Sts(\epsilon)) (\forall C_i \in C) : \exists t(x_{jt} > x_{it}). \quad (\text{A.36})$$

The above two expressions together imply that,

$$(\forall C_j \in S(Sts(\epsilon)) (\forall C_i \in C) : \exists t(x_{jt} > x_{it}). \quad (\text{A.37})$$

This means that no survivor of the filter could have been dominated by another alternative, or that for the bounded noise model, the survivor set of the Strongly Strict filter contains only true positives, i.e, no false positives. In other words, the Strongly Strict filter produces the Necessary set.

This completes the proof.

Theorem 6.6

Each alternative in the Sufficient Set satisfies the following condition with respect to every other alternative:

$$(\forall C_i \in PS) (\forall C_j \in C) \exists k(\tilde{x}_{ik} + \epsilon \geq \tilde{x}_{jk}). \quad (\text{A.38})$$

Proof:

Suppose that $C_i \in PS(\epsilon)$; also suppose there is another alternative C_j such that the following condition does not hold for C_i . In other words,

$$\forall k(\tilde{x}_{ik} + \epsilon < \tilde{x}_{jk}).$$

But from the definition of Superstrict dominance, this will imply that alternative C_j superstrictly-dominates C_i . Therefore C_i cannot belong to the sufficient set, $PS(\epsilon)$. So, we get a contradiction and the above condition must hold for each alternative in the sufficient set.

This completes the proof.

BIBLIOGRAPHY

- [1] Simon H. A. *The sciences of the artificial*. MIT Press, 1981.
- [2] Roy B. How outranking relation helps multiple criteria decision making. In J. L. Cochrane and M. Zeleny, editors, *Multiple Criteria Decision Making*, pages 179–201.
- [3] Roy B. The outranking approach and the foundations of electre method. In C. A. Bana e Costa, editor, *Readings in Multiple Criteria Decision Aids*, pages 155–183. Springer-Verlag, Heidelberg.
- [4] Roy B. Problems and methods with multiple objective functions. In *Mathematical Programming*, volume 1. 1971.
- [5] Roy B. Partial preference analysis and decision aids: The fuzzy outranking relation concept. In D. E. Bell, R.L. Keeney, and H. Raiffa, editors, *Conflicting Objectives in Decisions*, pages 40–75. 1977.
- [6] Roy B. Meaning and validity of interactive procedures as tools for decision making. *European Journal of Operational Research*, 31:297–303, 1987.
- [7] Roy B. and Mousseau V. A theoretical framework for analysing the notion of relative importance of criteria. *Journal of Multi-Criteria Decision Analysis*, 5(2):145–159, 1995.
- [8] Fishhoff Baruch. Value elicitation: Is anything in there? *American Psychologist*, 46:835–847, 1991.
- [9] J. L. Bentley, H. T. Kung, M. Schkolnick, and C. D. Thompson. On the average number of maxima in a set of vectors and applications. *Journal for the Association for Computing Machinery*, 25(4):536–543, 1978.
- [10] Fishburn P. C. Lexicographic orders, utilities and decision rules: A survey. *Management Science*, 20(11):1442–1471, 1974.

- [11] Pomerol J. C. Artificial intelligence and human decision making. *European Journal of Operational Research*, 99:3–25, 1997.
- [12] H. C. Calpine and A. Golding. Some properties of pareto optimal choices in decision problems. *International Journal of Management Science*, 4(2):141–147, 1976.
- [13] B. Chandrasekaran. Functional and diagrammatic representation for device libraries. Technical report, The Ohio State University, April 2000.
- [14] Hwang C.L. and Yoon K. *Multiple attribute decision making - methods and applications: A state-of-the-art-survey*. Springer-Verlag, New York, 1981.
- [15] I. Das and J. E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14(1):63–69, 1997.
- [16] Frisch Deborah and Robert T. Clemen. Beyond expected utility: Rethinking behavioral decision research. *Psychological Bulletin*, 116:46–54, 1994.
- [17] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.
- [18] Shafir E and Amos Tversky. Reason-based choice. *Cognition*, 49:11–36, 1993.
- [19] Steuer R. E. *Multiple Criteria Optimization: Theory, Computation and Applications*, pages 198–199. John Wiley and Sons, Inc, 1986.
- [20] Hobbs B. F. What can we learn from experiments in multiobjective decision analysis? *IEEE Transactions on Systems, Man, and Cybernetics*, 16(3):384–394, 1986.
- [21] Luce Mary Frances, John W. Payne, and James R. Bettman. Emotional trade-off difficulty and choice. *Journal of Marketing Research*, 36:143–159, 1999.
- [22] Nakayama H. On the components in interactive multiobjective programming methods. In *Plural Rationality and Interactive Decision Processes*, volume 248 of *Lecture Notes in Economics and Mathematical Systems*, pages 234–247. Springer-Verlag, 1985.
- [23] Nakayama H., Kanashige K., Takemoto S., and Watada Y. An application of a multi-objective programming technique to construction accuracy control of cable-stayed bridges. *European Journal of Operational Research*, 87(3):731–738, 1995.

- [24] Wallenius J. Comparative evaluation of some interactive approaches to multicriterion optimization. *Management Science*, 21(12):1387–1396, 1975.
- [25] Dyer J.S, Fishburn P.C., Steuer R.E., Wallenius J, and Zionts S. Multicriteria decision making, multiattribute utility theory: The next ten years. *Management Science*, 38(5):645–654, 1992.
- [26] MacCrimmon K.R. An overview of multiple objective decision making. In J.L. Cochrane and M. Zeleny, editors, *Multiple Criteria Decision Making*, pages 18–44. 1973.
- [27] H. T. Kung, F. Luccio, and F. P. Preparata. On finding the maxima of a set of vectors. *Journal of the Association for Computing Machinery*, 22(4):469–476, 1975.
- [28] Haim Levy. *Stochastic Dominance: Investment Decision Making under Uncertainty*. Kluwer Academic Publisher's, 1998.
- [29] Gardiner L.R. and Vanderpooten D. *Multicriteria Analysis*, chapter Interactive Multiple Criteria Procedures: Some Reflections, pages 290–301. Springer-Verlag, Heidelberg, 1997.
- [30] Despontin M. and Spronk J. *MCDM WorldScan*, chapter MCDM methods and software: a user-oriented catalog. 1988.
- [31] Hammersley J. M and Handscomb D. C. *Monte Carlo Methods*. John Wiley and Sons Inc, 1964.
- [32] J.M. Martel, N.T. Khoury, and M. Bergeron. An application of a multicriteria approach to portfolio comparisons. *Journal of Operational Research Society*, 39(7):617–628, 1988.
- [33] K. M. Miettinen. *Nonlinear Multiobjective Optimization*. International Series in Operations Research and Management Science. Kluwer Academic Publisher's, 1999.
- [34] Vladimir M. Ozeroy. Choosing the "best" multiple criteria decision-making method. *INFOR*, 30(2):159–171, May 1992.
- [35] Korhonen P. and Wallenius J. Behavioural issues in mcdm: Neglected research questions. *Journal of Multi-Criteria Decision Analysis*, 5(3):178–182, 1996.

- [36] Korhonen P. and Wallenius J. *Multicriteria Analysis*, chapter Behavioural Issues in MCDM: Neglected Research Questions, pages 412–422. Springer-Verlag, Heidelberg, 1997.
- [37] Wierzbicki A. P. A methodological approach to comparing parametric characterizations of efficient solutions. In *Large-Scale Modelling and Interactive Decision Analysis*, volume 448 of *Lecture Notes in Economics and Mathematical Systems*, pages 27–45. Springer-Verlag, 1986.
- [38] Slovic Paul. The construction of preference. *American Psychologist*, 50:364–371, 1995.
- [39] Slovic Paul and Lichtenstein S. Preference reversals: A broader perspective. *American Economic Review*, 73:596–605, 1983.
- [40] John W. Payne and James R. Bettman. Adaptive strategy selection in decision making. *Journal of Experimental Psychology*, 14(3):534–552, 1988.
- [41] John W. Payne, James R. Bettman, and David A. Schkade. Measuring constructed preferences: Towards a building code. *Journal of Risk and Uncertainty*, 19:243–270, 1999.
- [42] Josephson J. R., Chandrasekaran B., Carroll M., Iyer N. S., Wasacz B., Rizzoni G., Li Q., and Erb D. A. An architecture for exploring large design spaces. In *Proceedings of National Conference of the American Association for Artificial Intelligence, Madison, Wisconsin*, pages 143–150, 1998.
- [43] MacCrimmon K. R. Decision-making among multiple-attribute alternatives. Technical report, RAND, 1968.
- [44] MacCrimmon K. R. and J. K. Sin. Making trade-offs. *Decision Sciences*, 5:680–704, 1974.
- [45] Keeney R.L. and Raiffa H. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley and Sons, 1976.
- [46] French S. Interactive multi-objective programming: Its aims, applications and demands. *Journal of Operational Research*, 35(9):827–834, 1984.
- [47] Zionts S. *Multiple Criteria Decision Methods and Applications*, chapter Multiple criteria mathematical programming: An overview and several approaches, pages 85–128. Springer-Verlag, Heidelberg, 1985.

- [48] Zionts S. The state of multiple criteria decision making: Past, present, and the future. In A. Goicoechea, L. Duckstein, and S. Zionts, editors, *Multiple Criteria Decision Making: Proceedings of the Ninth International Conference: Theory and Applications in Business Industry, and Government*, pages 33–43. Springer-Verlag, New York, 1992.
- [49] Zionts S. Decision making: Some experiences, myths and observations. In *Multiple Criteria Decision Making: Proceedings of the Twelfth International Conference, Hagen (Germany)*, volume 448 of *Lecture Notes in Economics and Mathematical Systems*, pages 233–241. Springer-Verlag, Berlin, Heidelberg, 1997.
- [50] H. A. Simon. A behavioral model of rational choice. *Quarterly Journal of Economics*, 69(1):99–118, 1955.
- [51] Paul Slovic. Choice between equally valued alternatives. *Journal of Experimental Psychology*, 1(3):280–287, 1975.
- [52] W. Thorngate. Efficient decision heuristics. *Behavioral Science*, 25:219–225, 1980.
- [53] Amos Tversky. Contrasting rational and psychological principles in choice. In Richard J. Zeckhauser, Ralph L. Keeney, and James Sebenius, editors, *Wise Choices: Decisions, Games, and Negotiations*. Boston: Harvard Business School Press.
- [54] Amos Tversky. Intransitivity of preferences. *Psychological Review*, 76(1):31–48, 1969.
- [55] Amos Tversky. Elimination by aspects: a theory of choice. *Psychological Review*, 79:281–290, 1972.
- [56] Amos Tversky and Simonson I. Context-dependent preferences. *Management Science*, 39:1179–1189, 1993.
- [57] Amos Tversky, Slovic Paul, and Kahneman D. The causes of preference reversal. *American Economic review*, 4:201–211, 1990.
- [58] Siskos Y. and Spyridakos A. Intelligent multicriteria decision support: Overview and perspectives. *European Journal of Operational Research*, 113:236–246, 1999.
- [59] Siskos Y., Spyridakos A., and Yannacopoulos D. Using artificial intelligence and visual techniques into preference disaggregation analysis: The miidas system. *European Journal of Operational Research*, 113:281–299, 1999.