A New Solution Methodology for Min-Max Regret Robust Optimization for Interval Data Uncertainty using Priority Based Approximation Algorithms and Benders' Decomposition

A dissertation presented to the
Faculty of the Department of Industrial Engineering

in Partial Fulfillment of the Requirements for the Degree
Doctor of Philosophy
in Industrial Engineering

by

Ronny John George

December 2007

UMI Number: 3300011

A New Solution Methodology for Min-Max Regret Robust Optimization for Interval Data Uncertainty using Priority Based Approximation Algorithms and Benders' Decomposition

Ronny John George

Approved:

Chairman of the Committee
Dr. Tiravat Assavapokee
Assistant Professor
Industrial Engineering

Committee Members:

Dr. Hamid R. Parsaei
Professor & Chairman
Industrial Engineering

Dr. Maher Lahmar
Assistant Professor
Industrial Engineering

Dr. Jagannatha R. Rao
Associate Professor
Mechanical Engineering

Dr. Basheer Khumawala
Professor
Decision and Information Sciences

Larry C. Witte, Associate Dean
Cullen College of Engineering

Dr. Hamid R. Parsaei
Professor & Chairman
Industrial Engineering

Acknowledgements

I would like take this opportunity to thank the various people in my life that have helped and supported me during this endeavor. A dissertation is a significant undertaking and it is not possible without help and support from family, friends and my mentors.

Psalm 111:10 says "The fear of the Lord is the beginning of wisdom. A good understanding has all those who do his commandments; His praise endures forever". My belief in the Lord Almighty has carried me through this undertaking.

First and foremost I would like to thank my wife Anita; she has been my strongest supporter through all these years. She has waited patiently for me to achieve all my educational goals while her career and aspirations took a backseat. She has consistently motivated and encouraged me when the going got tough. Anita, I am forever in your debt.

I would also like to thank my parents George and Mary who have instilled in me the values to do good and to try and be the best at everything I attempt. My in-laws Joseph and Mary have always encouraged and supported me all through my higher education. Thanks also go to our siblings, Robin and family (Sajani, Jonathan & Jason) and Ansa and family (Lesliechayan, Anouska and George) for all their help, support and encouragement throughout these years.

I would also like to take this opportunity to thank my advisor Dr. Tiravat Assavapokee for supervising my dissertation and for being an excellent mentor. He has been extremely patient in explaining concepts. I have learned a lot from him. I would also like to thank Dr. Parsaei, Dr. Lahmar, Dr. Rao and Dr. Khumawalla for agreeing to serve on my committee.

I would also like to express my sincere gratitude to Dr. Cecil Mathews and Sunita Mathews for their support and kind words of encouragement during my studies. I am also grateful to Mathai uncle and Ammini aunty who opened their home to me so that I never felt homesick in Houston. Last but not the least; the journey would have been incredibly difficult without our good friends, Anil, Nita & Shveta. They have been our family away from home and have constantly encouraged and kept us cheerful throughout these years.

A New Solution Methodology for Min-Max Regret Robust Optimization for Interval Data Uncertainty using Priority Based Approximation Algorithms and Benders' Decomposition

An Abstract

of a

Dissertation

Presented to

the Faculty of the Department of Industrial Engineering

University of Houston

in Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in Industrial Engineering

by

Ronny John George

December 2007

Abstract

In this research, a new solution methodology for two-stage decision making under uncertainty and ambiguity is presented by using the deviation robust (min-max regret) criterion where the structure of the first stage problem is a mixed integer (binary) linear programming model and the structure of the second stage problem is a linear programming (LP) model. In the structure of the problem considered, each uncertain parameter can, independently of other parameters' settings, take its value from a real compact interval with unknown probability distribution. This new algorithm can be very useful in making deviation robust decisions under ambiguity when the joint probability distributions of key parameters are unknown and the only information available to decision maker are the potential ranges of uncertain parameters. Decision making problems of this type are very difficult to solve in general with the large number of uncertain parameters. The proposed algorithm coordinates four mathematical stages to efficiently solve the overall optimization problem. The algorithm sequentially solves and updates a relaxation problem until both feasibility and optimality conditions are satisfied. The feasibility and optimality verification steps involve the use of bi-level programming, which coordinate a Stackelberg game between the decision environment and decision makers. The proposed algorithm also incorporates approximation algorithms; priority based procedures, and accelerated Benders' decomposition algorithms to efficiently solve the overall optimization problem. A number of applications of the proposed algorithm on supply chain network infrastructure design problems are also demonstrated.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Introduction

In the real world, one can not avoid facing the necessity of making crucial decisions at every juncture. These decisions can have either a short term impact or a long term impact on subsequent outcomes. The facility location problem is one good example of long-term decision problems that are affected by a high level of uncertainty (i.e., demand uncertainty and supply uncertainty). Examples of these decisions include setting up a hospital system or a new school system to serve a certain region of population or setting up a supply chain infrastructure to serve a certain region of a company's customers. In order to make good long-term decisions under uncertainty, we first have to understand different types of decision making problems. The decision making problem can be categorized into three broad categories. (Taha, 1997)

1. Decision-making under certainty where all the parameters are *deterministic*.
2. Decision-making under risk where parameters have some type of *probability distribution.*
3. Decision-making under uncertainty where the probability distribution associated with the parameter is either unknown or cannot be determined.

The deterministic approach either completely ignores uncertainty or uses historical data and trends to determine the most likely scenario. This approach will use one instance of the data (most likely) as input to a decision model to generate an optimal decision with the use of single or multiple objectives. The major drawback of the deterministic approach is that it ignores instances other than the most likely one.

If an instance occurs that is different from the most likely instance, the generated solutions will be sub-optimal or even infeasible as there is usually no adjustment possible once the initial decision has been made. This approach is, most of the time, unacceptable in situations where the initial decision impacts various subsequent processes in future scenarios. For example, initial decisions like locating a manufacturing plant cannot be changed at whim.

The second category is also known as the stochastic optimization approach. This approach uses a probabilistic model to incorporate the possibility of occurrence of several instances. The problem is in obtaining information on the probability value with which these instances might be realized. The decision model will then try to generate a decision that will maximize or minimize an expected performance measure where the expectation is taken over the assumed or known probability distributions. This model usually assumes statistical independence between the various uncertain factors that may actually exist in the input data. This approach requires an intensive effort to assign probabilities to various data instances (future scenarios). When the number of scenarios is large and decision environments have multiple interdependent uncertain factors, this will be a futile exercise. Decision makers also will have difficulty in assigning exact probabilities to future scenarios which are interdependent on several factors.

The failure of both deterministic and stochastic approaches to model uncertainty is in their inability to recognize that associated with every decision is a whole distribution of outcomes depending on what data scenario is actually realized, and thus any approach that evaluates decisions using only one data scenario, either the most likely, is bound to fail (Kouvelis and Yu, 1996). As a decision maker, we may be interested in having information about the whole distribution of outcomes and planning for the worst case scenario especially for decisions of a unique nature which may be encountered only once. This is achieved by taking into account the risk of poor system performance for some realizations of data scenarios rather than optimizing expected system performance over all potential scenarios or just performance of the most likely scenario. Thus the performance of a decision across all potentially realizable scenarios is important. The decision making process based on uncertain information is often evaluated later as if the actual scenario realization has been known in advance of the decision. In these situations, a decision maker is rightfully concerned not only with how a decision's performance varies with the various data scenario realizations. He is also concerned with the comparison of actual system performance under the decision made versus the optimal performance that could have been achieved if perfect information on the scenario realization had been available prior to the decision making process. Neither deterministic nor stochastic optimization approaches can capture such concerns.

In the third category of decision making, there is a lack of complete knowledge about the random state of nature, which is subsequently reflected as considerable input data uncertainty to the supporting decision model. This dissertation falls into this third category which is known as the robust optimization approach. Instead of searching for the best solution for the most likely scenario or the best long run average optimal solution (which may not be obtainable due to the ambiguity in the input data), this approach searches for a solution that performs well (reasonable objective function value) across all possible input scenarios without attempting to assign probabilities to any scenario. Many criteria can be used to select among robust decisions. These criteria are, for instance, *min-max* criterion (*absolute robust* criterion), *min-max regret* criterion (*deviation robust* criterion), and *min-max relative* *regret* criterion (*relative robust* criterion). This approach is appropriate in two situations. The first situation is when decision makers are interested in hedging against the risk of poor system performance for some realizations of data scenarios as opposed to optimizing expected system performance over all potential scenarios, or just performance of the most likely scenario. The second situation (ambiguity situation) is when decision makers do not have enough historical information for uncertain parameters in the problem. Because, in this case, the expected performance of the solution cannot be measured, the *robust optimization* approach would be more appropriate.

As previously stated, there are three classical criteria in the robust decision making approach (Kouvelis and Yu, 1996). The first one is the *min-max* criterion also referred to as absolute robust criterion under which the robust decision is one for which the lowest (highest) level of benefit (cost) taken across all possible future input data scenarios is as high(low) as possible. This approach results in conservative decisions, as it concentrates the decisions only on the worst instance. Another approach is the *min-max regret* criterion also referred to as deviation robust criterion where the objective is to find the long-term decision which minimizes the maximum regret value overall possible input scenarios. The regret value under a specific input scenario is the difference between the resulting benefit (cost) to the decision maker based on the robust long-term decision and the benefit (cost) from the optimal decision that the decision maker would have taken if it is known with certainty that this specific input scenario will happen. The third approach is known as the *min-max relative regret* criterion also referred to as relative robust criterion where the objective is to find the long-term decision which minimizes the maximum relative regret value overall possible input scenarios. The relative regret value under a specific input scenario is defined as one minus the ratio of the resulting benefit to the decision maker based on the robust long-term decision and the corresponding benefit from the optimal decision that the decision maker would have taken when certain that a specific input scenario was to happen. It is worth noting that the decision making processes based on the last two criteria are less conservative since they take into account the magnitude of missed opportunities of a decision by benchmarking its performance with the performance of the optimal "realized" decision. This proposed research employs the deviation robust criterion for decision making under uncertainty.

In many decision making problems, mixed integer (binary) linear programming (MILP) models can often be applied to find optimal solutions to these decision problems (e.g. network infrastructure design problem and facility location problem in supply chain). In many cases, decision makers are facing with long-term decisions (e.g. capacity decision and/or location decision) that have to be made before the realization of uncertain parameters (first stage decisions). After these first stage decisions are made and decision makers obtain more information on model parameters, the short term decisions (second stage decisions) are then made under the fixed setting of the first stage decisions. This type of decision making process is referred to as two-stage decision making process.

The *focus of this dissertation is the development of efficient algorithms* for solving and understanding the two-stage decision making problem under interval data uncertainty when the base model of the problem can appropriately be formulated as a MILP model by utilizing various types of approximation algorithms, priority based procedures, and accelerated Benders' decomposition algorithms. In this work, an assumption is made that all the first stage decision variables are binary variables and all the second stage decision variables are continuous variables. In the structure of problem considered, each uncertainty parameter can take its values from a real compact interval with unknown probability distribution. This new algorithm can be effectively used in making deviation robust decisions when the joint probability distributions of key parameters are unknown and the only information available to decision makers are the potential range of uncertain parameters. In the following subsection, we give the more precise problem statement of this research.

1.2 Problem Statement

In this work, we address the two-stage decision making problem under uncertainty (ambiguity), where the uncertainty appears in the values of key parameters of a mixed integer linear programming formulation is presented as $\max_{\vec{x},\vec{y}}\{\vec{c}^T\vec{x}+\vec{q}^T\vec{y}+|\ W_1\vec{y}\leq\vec{h}_1+T_1\vec{x},\ W_2\vec{y}=\vec{h}_2+T_2\vec{x},\ \vec{x}\in\{0,1\}^{|\vec{x}|},\vec{y}\geq\vec{0},\}$. In the model, let the vector $\vec{x}$ represent the first-stage decision setting that has to be made before the realization of uncertainty and let the vector $\vec{y}$ represent the second-stage decision setting that can be made after the realization of uncertainty. Let the vectors $\vec{c},\vec{h}_1,\vec{h}_2,\vec{q}$ and the matrices T_1, T_2, W_1, and W_2 represent parameters of the decision model.

In the problem considered, each uncertain parameter (except parameters $\vec{q},W_1,W_2$) can take its value from a real compact interval with unknown probability distribution independently of other parameters' settings. We assume that the model parameters W_1 and W_2 are deterministic and each element of the parameter $\vec{q}$ can independently take its value from a finite set of real numbers.

As there is a lack of complete knowledge about the probability distribution of uncertain parameters in the considered problem, decision makers are not able to search for the first-stage decision setting (long term decision) with the best long run average performance. Instead, decision makers are searching for the first-stage decision setting that performs well (reasonable objective function value) across all possible input scenarios without attempting to assign assumed probability distribution to any ambiguous parameter. This resulting first-stage decision setting is referred to as the robust decision setting. In this research, we develop a new optimization algorithm for assisting decision makers who search for the robust first-stage decision setting under the deviation robustness definition (min-max regret robust solution) defined in Kouvelis and Yu (1996).

Traditionally, a min-max regret robust solution can be obtained by solving a scenario-based extensive form model of the problem which is also a mixed integer (binary) linear programming model (explained in detail in Chapter 3). The size of this extensive form model grows rapidly with the number of scenarios used to represent uncertainty as does the required computation time to find optimal solutions. Unfortunately in our case, the considered problem contains infinite number of scenarios to which the extensive form model can not be applied. In addition, the direct application of the Benders' decomposition also does not work for a problem with infinite number of possible scenarios.

In this research, we develop a new min-max regret robust optimization algorithm for two-stage mixed integer (binary) linear programming problems under this structure of parametric uncertainty. The algorithm is designed explicitly to handle an infinite set of possible scenarios. The algorithm can determine the robust values of the first-stage decision variables when the only information available to decision makers at the time of making the first stage decisions are a real compact interval containing possible values for each uncertain parameter with unknown probability distribution. The algorithm sequentially solves and updates a relaxation problem until both feasibility and optimality conditions of the overall problem are satisfied. The feasibility and optimality verification steps involve the use of bi-level programming, which coordinates a Stackelberg game (Von Stackelberg, 1943) between the decision environment and decision makers which is explained in Chapter 3.

This dissertation expands on the pioneering work done by Assavapokee et al.(2004, 2008a and 2008b) in solving large-scale two-stage decision problems under the interval data uncertainty and under the full factorial scenario design of data uncertainty by using deviation robust criterion.

We propose three main algorithmic improvements to the initial algorithm (Assavapokee 2004, 2008b) that will significantly speed up the computation time required by the algorithm.

The *first improvement* is the application of accelerated Benders' decomposition algorithm for solving the relaxation problem of the extensive form model. Even though the algorithm intends to keep the size of the relaxation problem at reasonable size, the number of scenarios considered by the relaxation problem may become relatively large for some practical decision problems. As a consequence, an efficient solution methodology for solving the relaxation problem is required in order to maximize the efficiency of the overall optimization algorithm. Fortunately, the relaxation problem has a unique structure that is suitable for Benders' decomposition algorithm. We propose to develop the special type of Benders' decomposition referred to as Accelerated Benders' decomposition algorithm suitable for solving this relaxation problem. This accelerated Benders' decomposition algorithm generates both strong optimality and feasibility cuts for the master problem. This application of the accelerated Benders' decomposition algorithm will not only increase the efficiency of the algorithm in solving the relaxation problem per iteration but also increase the efficiency of the overall algorithm.

The *second and third improvements* are the application of the priority based procedure, and the use of approximation algorithms, in solving the bi-level mixed integer nonlinear optimization problem required for checking the optimality condition of the overall algorithm. Even though the work by Assavapokee (2004, 2008b) presented the model transformation procedure that can transform the required bi-level mixed integer nonlinear optimization structure into a single-level mixed integer linear programming structure with complementary slackness constraints which is much easier to solve, the required computation time for solving this mathematical model is still relatively expensive compared to other components in the overall optimization algorithm. The main idea of the second improvement is to apply approximation algorithms to approximately solve this required bi-level programming problem and to classify the potential of each candidate solution generated by the algorithm. Only the candidate solution with promisingly high potential will be given the attention from the overall algorithm in solving its associated bi-level program to optimality. On the other hand, the candidate solution with significantly low potential will be discarded by the algorithm and there is no requirement in solving its associated bi-level programming problem. This application of this priority screening procedure will definitely increase the efficiency of the overall algorithm.

1.3 Structure of Dissertation

The dissertation is organized as follows. We first summarize the literature review of the related topics and then describe the new robust optimization methodology. The developed methodology will be validated using demonstrative examples. Chapter 2 contains the review of literature relevant to this dissertation. The literature will be classified into four areas: robust optimization, bi-level programming, decomposition methodology, and approximation algorithms. Chapter 3 contains the detailed formulation and detailed explanation of mathematical programming models required by the algorithm. Chapter 4 presents the case study results of the proposed algorithm to demonstrative examples on supply chain infrastructure design problem under interval data uncertainty. Chapter 5 presents the conclusions, summary and future directions for the research.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

This chapter presents the literature that is relevant to the development of the work in this dissertation. For the sake of clarity this chapter has been divided into four major sections. These sections are Robust Optimization, Bi-level Programming, Decomposition and Approximation Algorithms. Section 2.2 reviews the literature in the area of Robust Optimization. Section 2.3 reviews the literature in the area of Bi-level Programming. Section 2.4 reviews the literature on Decomposition methodology such as Benders and Accelerated Benders. Finally, Section 2.5 presents literature on approximation algorithms that can be used to efficiently search for a good solution in the real compact interval.

2.2 Literature Review on Robust Optimization

In mathematical programming most of the models developed are assumed to have input data that is precisely known and can be fixed at some specific value. However, most practical problems of interest have parameters whose values are uncertain. These data uncertainties affect the quality and feasibility of the model. If the realized value of the data is significantly different from the assumed value, the optimal solution found using the assumed value may no longer be optimal or even feasible. In the study carried out by Ben-Tal and Nemirovski (2000) they suggest that for most practical applications of Linear Programming, a small uncertainty in the data can make the usual optimal solution useless practically. Therefore we must design optimization models that take into account uncertainty in the data. Such models are also known as robust models.

One of the main approaches to handle uncertainty is the robust optimization. The aim of this approach is to produce solutions that will have a reasonable objective function value under any likely input data scenario to the decision model over a pre-specified time period. One possible criterion of selecting robust decision is the min-max regret criterion. Here we calculate the "regret" associated with every input data scenario. The "regret" of a scenario is measured as the closeness between the optimal objective function value for that scenario and the objective function value of the chosen solution for that scenario. The min-max criterion is then applied to the regret values, so as to choose the decision with the least maximum regret. A solution to a mathematical program is robust with respect to optimality if it remains close to optimality for any input data scenario. Robust optimization models can be divided into two broad categories: regret models and variability models.

A robust approach to solving linear optimization problems with uncertain data was first proposed by Soyster (1973). He proposed a linear optimization model to construct a solution that is feasible for all data that belong in a convex set. The resulting model can be considered conservative in that optimality is reduced in order to ensure robustness. In other words, we will accept a suboptimal solution for the nominal values of the data in order to ensure that the solution remains feasible and near optimal when the data changes.

Kouvelis and Yu (1996) summarize the work done in min-max regret optimization up to 1997. They also provide justification for the min-max regret approach and various aspects of applying it in practice. They define "close" to the optimal solution in several different ways. They define two regret criteria for robustness. The *robust deviation decision* is the decision that exhibits the best worst-case deviation from optimality. In other words, the robust deviation solution is one that minimizes the maximum regret over all possible realizations of the parameters in the model. This is the robustness definition used in this dissertation. The robust relative decision is the decision that exhibits the best worst-case percentage deviation from optimality. Another definition presented is that of *absolute robustness*. Absolute robustness evaluates the objective function value in each scenario without reference to the best possible decision that could have been made in that scenario and thus defines a solution that minimizes the maximum total costs. This approach is appropriate for high risk or highly competitive environments where even the worst case must guarantee a certain level of performance.

The robust deviation measure is chosen in this dissertation because it incorporates more information in the solution than absolute robustness which is useful for practical problems. Also, robust deviation gives more importance to scenarios that tend to produce large objective values compared to the other two measures. The use of the relative robustness measure results in more opportunity lost compared to the robust deviation measure. This is because scenarios that would tend to have very small positive or negative objective functions tend to always dominate solutions using a relative robustness measure.

Kouvelis and Yu (1996) used the scenarios approach to capture uncertainty for determining robustness. This approach can also be found in the stochastic optimization literature. Scenarios are decided upon and weights are placed on the realization of the scenarios. The final solution must satisfy each scenario and minimize some objective based on the difference between the proposed solution and optimal solution. In this respect the concept is close to robustness approach used in this dissertation.

The deviation robust approach has been used by Ammons and Realff (1999) to solve a mixed integer linear programming model for the robust infrastructure design for carpet recycling problems. Newton (2000) introduced a continuous robust approach using the deviation robustness definition. He used the information from parameter possible ranges for making robust infrastructure decisions for the reverse logistics problems instead of using discrete scenarios to capture uncertainty. This approach has many limitations in that it cannot handle the uncertainty when any coefficient of a continuous variable in the model is random. This approach also requires the assumption that there always exists a feasible robust infrastructure solution for the problem, which is not always true in general.

Gutierrez, Kouvelis, and Kurawarwala (1996) apply a different robustness approach which requires a robust network design to be within p% of the optimal solution for any realizable scenarios. They achieve this by the addition of a constraint to the model to ensure robustness. The model is solved by modifying Benders' decomposition algorithm to use cuts from one master problem on all scenarios. An alternative definition of robustness is to find a near-optimal solution that is not overly sensitive to any specific realization of the uncertainty (Bai, Carpenter and Mulvey, 1997). The goal is to minimize expected cost (maximize expected profit) and to reduce the variability over all possible scenarios.

The above robust optimization models thus include a measure of variability rather than regret. Variability can be measured either by variance (Hodder and Dincer, 1986; Mulvey, Vanderbei and Zenios, 1995; Bok, Lee, and Park, 1998) or by standard deviation (Goetschalckx, et al., 2001), both of which make the objective function a nonlinear function. Other measures of variability, include the von Neumann-Morganstern expected utility function (Bai, Carpenter and Mulvey, 1997) and the upper partial mean (Ahmed and Sahinidis, 1998), to allow asymmetry, but these functions are often hard to compute. When coefficients in a model are uncertain, the functional constraints may not necessarily be satisfied for all scenarios and therefore, it is convenient to introduce additional variables called recourse variables that represent the slack or surplus in the functional constraints. These variables are included in the objective function as an infeasibility penalty (Mulvey, Vanderbei and Zenios, 1995; Yu and Li, 2000).

Mulvey, Vanderbei, and Zenios (1995) were the first to present robust optimization as the integration of goal programming formulations with a scenario-based description of the problem data. In their work solution robustness is defined as the case when the optimal overall solution is near optimal for every possible demand scenarios and model robustness when the optimal overall solution is almost feasible for all scenarios. Solution robustness is achieved by the addition of variance or utility functions, to the objective function. A feasibility penalty function which is a function of the demand slack is added to the objective function to encourage model robustness. A penalty is assessed when the slack holds the positive or negative value, so the penalty applies when the model is infeasible, and when there is excess capacity.

Bok, Lee, and Park (1998) defined a quadratic objective function to maximize the expected net profit with penalties for the expected deviation of profit and excess capacity where the scenarios consist of different demand levels, each with an associated probability. The two-stage stochastic program is solved using Benders' decomposition methodology. Ahmed and Sahinidis (1998) use the definition of robustness of Mulvey, Vanderbei, and Zenios (1995), but propose alternative formulations to the mean plus variance objective function.

Ben-Tal and Nemirovski (1998, 1999, and 2000) address the robust solutions (min-max/max-min objective) by allowing the uncertainty sets for the data to be ellipsoids, and propose efficient algorithms to solve convex optimization problems under data uncertainty. However, as the resulting robust formulations involve conic quadratic problems, such methods cannot be directly applied to discrete optimization. Ben- Tal and Nemirovski (1998) study convex optimization problems for which the data is not specified exactly and is only known to belong to a given uncertainty set *U*, yet the constraints must hold for all possible values of the data from *U*. This works lays the foundation of robust convex optimization. They also show that if *U* is an ellipsoidal uncertainty set, then for some of the most important generic convex optimization problems (linear programming, quadratically constrained programming, semi definite programming and others) the corresponding robust convex program is either exactly, or approximately, a tractable problem which lends itself to efficient algorithms such as polynomial time interior point methods. Ben-Tal and Nemirovski (2000) also address Linear Programming (LP) problems with uncertain data. The focus is on uncertainty associated with hard constraints: those which must be satisfied, irrespective of the actual realization of the data (within a prescribed uncertainty set). They suggest a modeling methodology in which an uncertain LP is replaced by its Robust Counterpart (RC). They then develop the analytical and computational optimization tools to obtain robust solutions of an uncertain LP problem via solving the corresponding explicitly stated convex RC program. They show that the RC of an LP with ellipsoidal uncertainty set is computationally tractable, since it leads to a conic quadratic program, which can be solved in polynomial time.

Averbakh (2001) showed that polynomial solvability is preserved for a specific discrete optimization problem (selecting p elements of minimum total weight out of a set of m elements with uncertainty in weights of the elements) when each weight can vary within an interval under the min-max-regret robustness. However, the approach does not seem to generalize to other discrete optimization problems.

Bertsimas and Sim (2003) propose an approach to address data uncertainty for discrete optimization and network flow problems that allows controlling the degree of conservatism of the solution, and is computationally tractable both practically and theoretically. They propose a robust integer programming problem of moderately larger size that allows controlling the degree of conservatism of the solution in terms of probabilistic bounds on constraint violation. This approach is useful when both the cost coefficients and the data in the constraints of an integer programming problem are subject to uncertainty. When only the cost coefficients are subject to uncertainty and the problem is a $0 - 1$ discrete optimization problem on n variables, then they solve the robust counterpart by solving at most $n + 1$ instances of the original problem. Thus, the robust counterpart of a polynomial solvable $0 - 1$ discrete optimization problem remains polynomial solvable. Thus robust matching, spanning tree, shortest path, matroid intersection, etc. are polynomial solvable. They also show that the robust counterpart of an NP-hard α-approximable $0 - 1$ discrete optimization problem, remains α-approximable. Finally, they propose an algorithm for robust network flows that solves the robust counterpart by solving a polynomial number of nominal minimum cost flow problems in a modified network.

Bertsimas and Sim (2004) present a new robust approach to solve linear optimization problems with uncertain data. They propose an approach that makes the tradeoff between optimality and feasibility more attractive. They thus try to reduce the "price of robustness". In particular they adjust the level of conservatism of the robust solutions in terms of probabilistic bounds of constraint violations. They propose a robust formulation that is linear, is able to withstand parameter uncertainty under the model of data uncertainty U without excessively affecting the objective function, and readily extends to discrete optimization problems. Their method is as follows. Consider the i^{th} constraint of the nominal problem $a_i'x \le b_i$. Let J_i be the set of coefficients $a_{ij}, j \in J_i$ that are subject to parameter uncertainty. For every i they introduce a parameter Γ_i, not necessarily integer that takes values in the interval $\left[0, |J_i|\right]$. The role of the parameter Γ_i is to adjust the robustness of the proposed method against the level of conservatism of the solution. The goal is to be protected against all cases that up to $\lfloor \Gamma_i \rfloor$ of the coefficients $a_{ij}, j \in J_i$ are allowed to change and one coefficient a_{ij} changes by $\left(\Gamma_i - \lfloor \Gamma_i \rfloor\right)\hat{a}_{it}$. They stipulate that nature will be restricted in its behavior, in that only a subset of the coefficients will change in order to adversely affect the solution. They thus develop an approach that has the property that if nature behaves like this, then the robust solution will be feasible *deterministically* and moreover even if more than $\lfloor \Gamma_i \rfloor$ change, then the robust solution will be feasible with *very high probability*.

Butler (2003) proposes a new definition of a robust solution by combining the expected value and the relative robustness definition for an application of supply chain design for new product distribution. Montemanni, R. and Gambardella,L.M. (2002) apply the relative robustness criterion to the shortest path problem defined on a directed graph G = (V,A), where V is a set of vertices and A is a set of arcs. A starting vertex $s \in V$, and a destination vertex $t \in V$ are given and an interval $\left[l_{ij}, u_{ij}\right]$, with $u_{ij} \ge l_{ij} > 0$, is associated with each arc $(i, j) \in A$. Intervals represent ranges of possible costs. They model uncertainty about the exact value of these costs. Their work is concerned with transport problems, and for this reason each $\left[l_{ij}, u_{ij}\right]$ is an interval of possible travel times for the road associated with arc (i, j).

Assavapokee (2004, 2008a) present a scenario relaxation (SR) algorithm for solving large-scale general scenario-based min-max regret and min-max relative regret robust optimization problems for two-stage mixed integer linear programming formulations. Results are reported showing the significant improvement in computation time of the scenario relaxation algorithm over the extensive form method. They also develop two new robust optimization algorithms under deviation robust criterion for two-stage decision making under interval data uncertainty (Assavapokee 2004, 2008b) and under full-factorial scenario design (Assavapokee et al., 2007a) for the mixed integer (binary) linear programming formulation. In Assavapokee (2004, 2008b), they present a min-max regret robust optimization algorithm for two stage decision making under uncertainty where the structure of the first stage problem is a mixed integer linear programming model and the structure of the second stage problem is a linear programming model. In the structure of problem considered, the parametric uncertainty is represented by real compact intervals. Their algorithm can be effectively used in making min-max regret robust decisions when the joint probability distributions of key parameters are unknown and the only information available to decision maker are the potential ranges of uncertain parameters. The algorithm coordinates three mathematical programming formulations to solve the overall optimization problem. They also provide a counter-example that illustrates the insufficiency of the robust solution obtained by only considering a finite number of scenarios generated by end points of all compact intervals. They also demonstrate the application of the algorithm on a supply chain network infrastructure design problem.

Assavapokee et al., (2007a) present a modified min-max regret robust optimization algorithm where each uncertain parameter can, independently of other parameters' setting, take its value from a finite set of real numbers with unknown probability distribution. This structure of parametric uncertainty is referred to as the full-factorial scenario design of data uncertainty. The proposed algorithm is shown to be very effective for solving large-scale scenario based min-max regret robust optimization problems under this structure of parametric uncertainty. The algorithm coordinates three mathematical programming formulations to effectively solve the overall optimization problem.

Assavapokee et al., (2007b) modify the previous algorithm to solve the problem under the relative robust criterion. The algorithm also coordinates three computational stages to effectively solve the overall optimization problem. Bi-level programming formulations and fractional programming concepts are the main components of the proposed algorithm.

2.3 Literature Review on Bi-level Programming

The bi-level programming problem (*BLPP*) is a static variation of the problem introduced by Von Stackelberg (1952) in the study of unbalanced economic markets. In the basic model, control of the decision variables is partitioned amongst the players who seek to optimize their individual payoff functions. Perfect information is assumed so that both players know the objective and feasible choices available to the other. The leader goes first and attempts to optimize his objective function. In doing so he must anticipate all possible responses of his opponent, termed the follower. The follower observes the leader's decision and reacts in a way that is personally optimal without regard to external effects. Because the set of feasible choices available to either player is interdependent, the leader's actions affects both the follower's payoff and allowable actions and vice versa. The fact that the game is said to be 'static' implies that each player has only one move. The vast majority of research on bi-level programming has centered on the linear version of the problem, alternatively known as the linear Stackelberg game (Bard, 1999).

The general structure of a bi-level programming problem has been put forward by Bard (1999). Suppose that the higher-level decision maker or leader has control over the vector $x \in X \subseteq R^n$ and that the subunits, collectively called the follower, have control over the vector $y \in Y \subseteq R^m$. The leader goes first and selects an x in an attempt to minimize $F(x, y(x))$ subject perhaps to some additional constraints. The notation $y(x)$ stresses the fact that the leader's problem is implicit in the y variables. The follower observes the leader's actions and reacts by selecting a y to minimize the objective function $f(x, y)$, subject to a set of constraints in the y variables for the particular value of x chosen. When the feasible region of either player can be described by inequality constraints, the general bi-level programming is written as

$$For\ x \in X \subset R^n, y \in Y \subset R^m, F : X \times Y \to R^1, and\ f : X \times Y \to R^1$$

$$\left.\begin{array}{l} \min\limits_{x \in X} F(x, y) = c_1 x + d_1 y \\ subject\ \ to \quad A_1 x + B_1 y \le b_1 \end{array}\right\} Leader's\ objective\ function$$

$$\left.\begin{array}{l} \min\limits_{y \in Y} f(x, y) = c_2 x + d_2 y \\ \qquad\qquad subject\ \ to \quad A_2 x + B_2 y \le b_2 \end{array}\right\} Follower's\ objective\ function$$

$$where\ c_1, c_2 \in R^n, d_1, d_2 \in R^m, b_1 \in R^p, b_2 \in R^q, A_1 \in R^{p\times n}, B_1 \in R^{p\times m},$$
$$A_2 \in R^{q\times n}, and\ B_2 \in R^{q\times m}.$$

The sets X and Y place additional restrictions on the variables such as nonnegativity or integrality requirements. After the leader selects the x value, the first term in the follower's objective function becomes a constant and can be removed from the problem. Thus $f(x, y)$ can be replaced by $f(y)$. Thus y can be viewed as a function of x.

The definitions are used for the solution model of the *BLPP* model.

a. Constraint region of the BLPP;

$$S \triangleq \{(x, y) \mid x \in X, y \in Y, A_1 x + B_1 y \le b_1, A_2 x + B_2 y \le b_2\}$$

b. Feasible set for the follower for each fixed $\hat{x} \in X$:

$$S(\hat{x}) \triangleq \{y \in Y \mid A_2 \hat{x} + B_2 y \le b_2\}$$

c. Projection of S onto the leader's decision space:

$$S(X) \triangleq \{x \in X \mid \exists y \in Y, A_1 x + B_1 y \le b_1, A_2 x + B_2 y \le b_2\}$$

d. Follower's rational reaction set for $\hat{x} \in S(X)$:

$$P(\hat{x}) \triangleq \left\{y \in Y \mid y \in \arg\min\left[f(\hat{x}, \hat{y}) \mid \hat{y} \in S(\hat{x})\right]\right\}$$

e. Inducible region:

$$IR \triangleq \{(x, y) \mid (x, y) \in S, y \in P(x)\}$$

To ensure that the *BLPP* model is well posed, we assume that S is nonempty and compact; i.e. $P(X) \ne \phi$. The rational reaction set $P(x)$ defines the response while the inducible region (*IR*) represents the set over which the leader may optimize. The *BLPP* model can be written as $\min\{F(x, y) \mid (x, y) \in IR\}$. To have an explicit representation of the inducible region, we can use Karush-Kuhn-Tucker (KKT) conditions to rewrite the follower problem. The resultant *BLPP* model is

$$
\begin{aligned}
&\min F(x,y) = c_1x + d_1y \\
&\textit{subject to} \quad A_1x + B_1y \le b_1 \;\; \textit{and} \;\; A_2x + B_2y \le b_2 \\
&\qquad\qquad uB_2 - v = -d_2 \\
&\qquad\qquad u(b_2 - A_2x - B_2y) = 0 \;\; \textit{and} \;\; vy = 0 \\
&\qquad\qquad x \ge 0, y \ge 0, u \ge 0, v \ge 0 \\
&\textit{where} \; u \in R^q \;\; \textit{and} \;\; v \in R^m.
\end{aligned}
$$

Bard (1999) suggests the use of the Big M method with binary variables to handle nonlinear constraints (complementary slackness conditions) in this model. The drawbacks of this method however do come up in practical applications and is explained later when we discuss the three stage algorithm. This research applies bi-level programming in the second, third and fourth stages of the algorithm. Bi-level linear optimization was first proposed in the mid-1960's in the initial work by Baumol and Fabian (1964). The linear bi-level programming problem was first shown to be NP-hard by Jeroslow (1985). Bard (1991) provided an alternative proof by constructively reducing the problem of maximizing a strictly convex quadratic function over a polyhedron to a linear max-min problem.

In general, there are three different workable approaches for solving a linear bi-level programming problem. The first approach makes use of the theorem that the solution of the linear bi-level programming problem occurs at a vertex of S and involves some form of vertex enumeration in the context of the simplex method. Candler and Townsley (1982) were the first to develop an algorithm that was globally optimal. Their algorithm repeatedly solves two linear programs, one for the leader in all of the x variables and a subset of the y variables associated with an optimal basis to the follower's problem, and the other for the follower with all the x variables fixed. They thus explore optimal bases of the follower's problem for x fixed and then return to the leader's problem with the corresponding basic y variables. They are able to provide a monotonic decrease in the number of follower bases that have to be examined by focusing on the reduced cost coefficients of the y variables not in an optimal basis of the follower's problem.

The second approach for solving the linear bi-level programming problem known as the "Kuhn-Tucker" approach is to use a branch and bound strategy to deal with the complementarity constraints. Omitting or relaxing this constraint leaves a standard linear programming which is easy to solve. The various methods proposed employ different techniques for assuring that complementarity is ultimately satisfied (Bard and Moore, 1990; and Judice and Faustino, 1992).

The third method is based on some form of penalty approach. Aiyoshi and Shimizu (1984) addressed the general bi-level programming problem by first converting the follower's problem to an unconstrained mathematical program using a barrier method. The corresponding stationary conditions are then appended to the leader's problem, which is solved repeatedly for decreasing values of the barrier parameter. To guarantee convergence the follower's objective function must be strictly convex. A different approach using an exterior penalty method was proposed by Shimizu and Lu (1995) that simply requires convexity of all the functions to guarantee global convergence.

Edmunds and Bard (1991) present two algorithms for solving various versions of the leader follower game when certain convexity conditions hold. They use a hybrid branch and bound scheme which does not guarantee global optimality for one algorithm. Another algorithm is based on objective function cuts and is guaranteed to converge to an ε-optimal solution, barring numerical stability problems with the optimizer. They also examine the performance of the two algorithms using randomly generated test problems.

Dempe et al., (2005) present a mathematical framework for the problem of minimizing the cash-out penalties of a natural gas shipper. The problem is modeled as a mixed-integer bi-level programming problem having one Boolean variable in the lower level problem. Such problems are difficult to solve. To obtain a more tractable problem they move the Boolean variable from the lower to the upper level problem. The implications of this change of the problem are investigated thoroughly. The resulting lower level problem is a generalized transportation problem. The corresponding results are then used to find a bound on the optimal function value of the initial problem.

Bjorndal and Jornsten (2005) present a bi-level programming formulation of a deregulated electricity market. They show that the relation of the deregulated electricity market to general economic models can be formulated as bi-level programming problems. They also provide an explanation of the reason why several theorems can be proven to be false for electricity networks. The interpretation of the deregulated electricity market as a bi-level program also indicates the magnitude of the error that can be made if the electricity market models studied do not take into account the physical constraints of the electric grid or oversimplification of the electricity network to a radial network.

Cao et al., (2006) present a capacitated plant selection problem in a decentralized manufacturing environment where the principal firm and the auxiliary plants operate independently in an organizational hierarchy. A non-monolithic model is developed for plant selection in the decentralized decision making process. The developed model considers the independence relationship between the principal firm and the selected plants. It also takes into account the opportunity costs of over-setting production capacities in the opened plants. The developed mathematical programming model is a two-level nonlinear programming model with integer and continuous decision variables that is transformed into an equivalent single level model, linearized and solved.

2.4 Literature Review on Decomposition Methods

In general, a decomposition principle is a systematic procedure for solving large-scale general mathematical programs or specific mathematical programs with special structure. The strategy of a decomposition procedure is to iterate between two separate mathematical programs. Information is passed back and forth until a point is reached where the solution to the original problem is achieved. It is not unusual for realistically sized mathematical models to produce mixed integer linear programs with many thousands or even millions of rows and columns. To solve such problems, some method must be applied to convert the large problems into one or more appropriately coordinated smaller problems of manageable size.

In some applications of linear programming the constraints of the problem can be divided into two groups, one group of "easy" constraints and another of "hard" constraints. This usually happens in network set up problems where the constraints that describe the network (the easy constraints) are augmented by additional constraints of a more general form (the hard constraints). This can also happen in "block angular" problems described below where there are a small number of constraints that involve all the variables (the hard constraints) but if these are removed the problem decomposes into several independent smaller problems, each of which is easier to solve. The "hard" constraints need not be in themselves intrinsically difficult, but rather they can complicate the linear program, making the overall problem more difficult to solve. If these "complicating" constraints could be removed from the problem, then more efficient techniques could be applied to solve the resulting linear program. The Decomposition Principle is a tool for solving linear programs having this structure. Popular decomposition methodologies include Dantzig-Wolfe decomposition (Dantzig and Wolfe, 1960), Benders' decomposition (Benders, 1962) and Lagrangian relaxation techniques (Falk, 1967). The Benders Decomposition technique is based on delayed constraint generation while Dantzig-Wolfe decomposition is based on delayed column generation.

To explain the concept in detail, let us consider the development of a master corporate plan for several production facilities. Although each facility has its own independent capacity and production constraints, the different facilities are tied together at the corporate level by budgetary constraints. The resulting model consists of two types of constraints: hard representing the corporate budgetary constraints; and soft representing the internal capacity and production restrictions of each facility. The layout of the constraints for ***n*** activities (facilities) is shown in Figure 2.1. In the absence of the common constraints, all the activities operate independently. The decomposition algorithm improves the computational efficiency of the problem by breaking it down into ***n*** sub problems that can be solved almost independently. The resulting "block angular structure" is observed in Figure 2.2.

$$\begin{array}{lllllll}
\textit{Maximize} & z = C_1X_1 + C_2X_2 + \ldots + C_nX_n & & & & & \\
\textit{subject to} & A_1X_1 + A_2X_2 + \ldots\ldots + A_nX_n & = b_0 \\
& D_1X_1 & = b_1 \\
& \qquad D_2X_2 & = b_2 \\
& \qquad\qquad . & . \\
& \qquad\qquad\quad . & . \\
& \qquad\qquad\qquad D_nX_n & = b_n \\
& X_j \geq 0, \;\; j = 1, 2, \ldots, n &
\end{array}$$

Figure 2.1: General Form of the Model

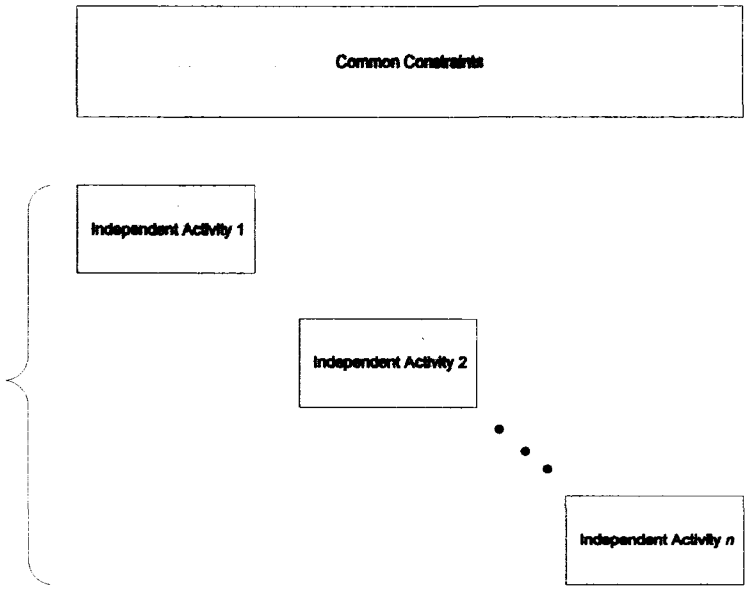


Figure 2.2: Block Angular Structure

Benders (1962) presented a classical solution approach to solve combinatorial optimization problems, based on the ideas of partition and delayed constraint generation. The method partitions the problem to be solved into two simpler problems, named *master* and *subproblem*. The master problem is a relaxed version of the original problem containing only a subset of the original variables and the associated constraints. The sub-problem is the original problem with the variables obtained in the master problem fixed and is used to generate cuts for the master problem. The master and subproblems are solved iteratively until no more cuts can be generated. This is usually obtained by the use of a ε-optimal termination criterion. The conjunction of the variables found in the last master and subproblem iteration is the solution to the original formulation. The decomposition approach is justified only if the master and subproblem approach can be solved efficiently. As the number of scenarios increases we would need to solve the master and subproblems several times. For a network design problem, the master problem could deal with integer variables that define the network while the subproblem works with the continuous variables representing the actual flow of commodities for the tentative network obtained in the master problem.

Geoffrion and Craves (1974) present a solution technique based on Benders Decomposition to solve the multicommodity capacitated single-period distribution system design problem of optimal location of intermediate distribution facilities between plants and customers. The model which is formulated as a mixed integer linear program was solved to optimality and proven with a small number of Benders cuts. They also provide a discussion on the reason this problem class appears to be so amenable to solution by Benders' method.

Magnanti and Wong (1981) propose a methodology for improving the performance of Benders decomposition when applied to mixed integer problems. They introduce a new technique for accelerating the convergence of the algorithm and theory for distinguishing "good" model formulations of a problem. The acceleration technique is based upon selecting from among the alternate optima of the Benders subproblem to generate strong or pareto-optimal cuts. The cut generation technique leads to very efficient algorithms that exploit underlying structure of the models especially for network location problems. They also suggest criteria for comparing various mixed integer formulations of a problem and to choose formulations that provide stronger cuts for Benders decomposition.

Magnanti et al., (1986) extended the study to the context of the uncapacitated network design problem. They show that the generality of Benders decomposition by proving that some known cuts are actually Benders cuts for particular choices of the integer variables. The master problem proposes a tentative network by setting the integer variables and the subproblem finds the continuous variables flow distribution. They show that for this particular case, Pareto-optimal cuts can be generated at the price of solving *k minimum cost flow problems* (one for each commodity). They also advocate the use of Benders decomposition in conjunction with other approaches.

Gutierrez et al., (1996) extended the work of Magnanti et al., (1986). They proposed a robust approach able to consider the uncertainty in the transportation costs and in the demands. The data uncertainty is described through a set of scenarios, each with different values for the transportation costs. The solution of the model is done by the use of a multi-master Benders algorithm, where an individual master problem is associated with each possible scenario. Each time a master problem is solved there is a cross generation of Benders cuts, i.e. the subproblem generates a cut for each of the master problems. They conclude that this cross generation accelerates the convergence of the algorithm.

Randazzo and Luna (2001) present a comparison of some optimal methods applied to a problem of local access network design. They define two equivalent flow formulations for the problem, the first a single commodity and the second being a multicommodity flow model. The objective in both cases is the cost minimization of the sum of the fixed (structural) and variable (operational) costs of all the arcs composing an arborescence that links the origin node to every demand node. The weak single commodity flow formulation is solved by a branch-and-bound strategy that applies Lagrangian relaxation for computing the bounds. The strong multicommodity flow model is solved by a branch-and-cut algorithm and by Benders decomposition. Their experience suggests that a certain number of the modeling and solution strategies can be applied to frequently occurring problems where basic optimal solutions to the linear program are automatically integral which also solves the combinatorial optimization problem. They conclude that a well tailored Benders partitioning approach emerges as a robust method to cope with the fabricated cases where the linear programming relaxation exhibits a gap between the continuous and the integral optimal values. The most important conclusion is the fact that Benders decomposition although slower than the branch- and-cut on six of the 30 instances was the only algorithm able to solve to optimality all instances within the limit time (24 hours).

The decomposition methodology that we use to solve one of the mathematical models (section 3.5.1) in this dissertation is an accelerated Benders' decomposition algorithm (Santoso, 2005). This is possible because of the special block angular structure of this specific mathematical model. This specific model contains only one set of binary decision variables for all input scenarios. If their values can be fixed, the problem can be partitioned into several linear programming problems (one for each scenario) that can be solved independently. For this reason, this specific mathematical model is an ideal problem structure for applying the Accelerated Benders' decomposition algorithm.

Geoffrion (1972) generalized the Bender's approach to a broader class of problems in which the parameterized subproblem is not constrained to be a linear program. Nonlinear convex duality theory was employed to derive the natural families of cuts corresponding to those in Bender's case. The main result is an extension of Bender's approach to a more general class of problems with the help of nonlinear duality theory. The problems are of the type

$$\underset{x,y}{\text{maximize}}\, f(x,y) \textit{ subject to } G(x,y) \geq 0, x \in X, y \in Y \qquad (2.4.1)$$

where y is a vector of hard variables in the sense that eq. 2.4.1 is a much easier optimization problem in x when y is temporarily held fixed. G is an m-vector of constraint functions defined on $X \times Y \subset R^p \times R^q$. The situations modeled are of the type:

a. For fixed y, eq. 2.4.1 separates into a number of independent optimization problems, each involving a different subvector of x.

b. For fixed y, eq. 2.4.1 assumes a well-known special structure (block angular structure) for which efficient solution procedures are available; and

c. Equation 2.4.1 is not a concave program in x and y jointly, but fixing y renders it so in x.

Geoffrion postulated that it is easier to obtain computational economies by looking at problem eq. 2.4.1 in y space instead of the xy-space. The key idea to achieve this is the concept of projection, also known as partitioning. The projection of eq. 2.4.1 onto y is

$$\underset{y}{\text{maximize}}\, v(y) \text{ subject to } y \in Y \cap V \tag{2.4.2}$$

where

$$v(y) \equiv \underset{x}{\text{supremum}}\, f(x,y) \text{ subject to } G(x,y) \geq 0, x \in X \tag{2.4.3}$$

and

$$V \equiv \left\{y : G(x,y) \geq 0 \ \text{ for some } x \in X\right\}. \tag{2.4.4}$$

Here $v(y)$ is the optimal value of eq. 2.4.1 for fixed y and since y is designated as a hard variable, evaluating $v(y)$ is easier to solve that eq. 2.4.1. Equation 2.4.3 can also be represented as

$$\underset{x \in X}{\text{maximize}}\, f(x,y) \text{ subject to } G(x,y) \geq 0. \tag{2.4.5}$$

The set V consists of those values of y for which eq. 2.4.5 is feasible; $Y \cap V$ can be considered as the projection of the feasible region of eq. 2.4.1 onto y-space. They also define

$$L^{*}(y;u) \equiv \underset{x \in X}{\text{supremum}}\left\{f(x,y) + u^{t}G(x,y)\right\}, y \in Y, u \geq 0, \tag{2.4.6}$$

$$L_{*}(y;\lambda) \equiv \underset{x \in X}{\text{supremum}}\left\{\lambda^{t}G(x,y)\right\}, \qquad y \in Y, \lambda \geq 0. \tag{2.4.7}$$

The generalized Benders decomposition procedure can be summarized in the following steps:

1. Let a point $\bar{y} \in Y \cap V$ be known. Solve sub-problem for $\bar{y}$ using eq. 2.4.5 and obtain an optimal (or near-optimal) multiplier vector $\bar{u}$ and the function $L^{*}(y;\bar{u})$. Put p =1, q =0, $u^{1} = \bar{u}$, Lower Bound (LB)= $v(\bar{y})$.
2. Select the convergence termination parameter $\varepsilon > 0$.
3. Solve the current relaxed *master* problem

$$\underset{\substack{y \in Y \\ y_0}}{\text{maximize}}\, y_0 \text{ subject to } y_0 \leq L^{*}(y;u^{j}), \qquad j = 1,2,......,p,$$

$$L_{*}(y;\lambda^{j}) \geq 0, \qquad j = 1,2,.......,q,$$

by any applicable algorithm. Let $(\hat{y}, \widehat{y_0})$ be an optimal solution, $\widehat{y_0}$ is an upper bound (UB) on the optimal value of eq. 2.4.1. If LB = $\widehat{y_0} - \varepsilon$, terminate.

4. Solve the revised subproblem for $\hat{y}$. Either the quantity $v(\hat{y})$ is finite. If $v(\hat{y}) \geq \widehat{y_0} - \varepsilon$ terminate. Otherwise determine an optimal multiplier vector $\hat{u}$. If none exists a near optimal multiplier vector satisfying $\widehat{y_0} > \underset{x \in X}{\text{supremum}} \left\{ f(x,\hat{y}) + \hat{u}^{t} G(x,\hat{y}) \right\}$ will suffice. Determine the function $L^{*}(y;\hat{u})$. Increase p by 1 and put $u^{p} = \hat{u}$. If $v(\hat{y}) > LB$, put $LB = v(\hat{y})$. LB is a lower bound on the optimal value of eq. 2.4.1. Return to step 4.
5. If 2.4.5 is infeasible for $\hat{y}$ determine $\hat{\lambda}$ *in* Λ satisfying $\underset{x \in X}{\text{supremum}} \left\{ (\hat{\lambda}^{t} G(x,\hat{y}) \right\} < 0$. Determine the function $L_{*}(y;\hat{\lambda})$. Increase q by 1 and put $\lambda^{q} = \hat{\lambda}$. Return to step 3.

2.5 Literature Review on Approximation Algorithms

Heuristic algorithms (including artificial intelligence based heuristics, simulated annealing and genetic algorithms) have been used to tackle discrete optimization problems in recent years. Bi-level programming problems inherit many structural properties from discrete optimization at least if the lower level problem is replaced by it's (under convexity and regularity assumptions equivalent) Karush-Kuhn-Tucker conditions. The use of heuristic algorithms to solve bi-level programming problems has been done in the research Anandalingam et al., (1983), Friesz et al., (1992,1993), Gendreau et al., (1996) and Marcotte (1986).

Only a very few results are available at the moment with respect to algorithms solving mixed-discrete bi-level programming problems. Such algorithms are the following: branch-and-bound algorithms for exact and approximate solutions (Edmunds and Bard, 1992), (Bard and Moore, 1990), and (Wen and Yang, 1990), algorithms being based on cutting plane algorithm (Dempe, 1996b) and the k-th best algorithm (Thirwani and Arora, 1997). Other algorithms include trust region methods of (Marcotte et al., 2001), (Scholtes and Stohr, 1999).

2.6 Summary

In this chapter, we summarize the work done in the areas of Robust Optimization, Bi-level Programming, Decomposition methods and search algorithms for Bi-level programming. In the next chapter we explain the existing three-stage algorithm developed by Assavapokee (2004, 2008b) and its drawbacks. We then propose a modified four stage algorithm that eliminates these drawbacks with the help of new solution techniques utilizing Accelerated Benders Decomposition methods, priority based solution procedure, and approximation algorithms.

Chapter 3

METHODOLOGY

3.1 Introduction

This chapter begins by reviewing key concepts of scenario based min-max regret robust optimization (i.e., types of decisions and extensive form formulation). We then explain the concept of the two-stage decision making process and give a summary of the three-stage deviation robust algorithm developed by Assavapokee (2004, 2008b). We also discuss the inherent drawbacks of the algorithm. Next the proposed four-stage deviation robust algorithm is presented that improves the solution time by augmenting the existing three-stage deviation robust algorithm by the use of approximation and decomposition algorithms. The methodology of the new algorithm is then summarized and explained in detail, and each of its four stages is specified. The chapter concludes with the theoretical result that the algorithm always terminates at the min-max regret robust optimal solution (if one exists) in a finite number of iterations. The case studies to validate the algorithm are presented in Chapter 4.

3.2 Deviation Robust Optimization Models

We address the problem where the basic components of the model's uncertainty are represented by a finite set of all possible scenarios of input parameters, referred as the scenario set $\bar{\Omega}$. The problem contains two types of decision variables. The first stage variables model binary choice decisions, which have to be made before the realization of uncertainty. The second stage decisions are continuous recourse decisions, which can be made after the realization of uncertainty. Let vector $\vec{x}_\omega$

denote binary choice decision variables and let vector $\vec{y}_\omega$ denote continuous recourse decision variables and let $\vec{c}_\omega, \vec{q}_\omega, \vec{h}_{1\omega}, \vec{h}_{2\omega}$ $, W_{1\omega}, W_{2\omega}, T_{1\omega}$, and $T_{2\omega}$ denote model parameters setting for each scenario $\omega \in \bar{\Omega}$. If the realization of model parameters is known to be scenario ω a priori, the optimal choice for the decision variables $\vec{x}_\omega$ and $\vec{y}_\omega$ can be obtained by solving the following model (1):

$$O_\omega^* = \left\{ \begin{array}{ll} \max\limits_{\vec{x}_\omega, \vec{y}_\omega} c_\omega{}^T \vec{x}_\omega + \vec{q}_\omega{}^T \vec{y}_\omega & \\ s.t. \quad W_{1\omega}\vec{y}_\omega - T_{1\omega}\vec{x}_\omega \le \vec{h}_{1\omega} & \\ \quad\quad W_{2\omega}\vec{y}_\omega - T_{2\omega}\vec{x}_\omega = \vec{h}_{2\omega} & \\ \quad\quad \vec{x}_\omega \in \{0,1\}^{|\vec{x}_\omega|} \text{ and } \vec{y}_\omega \ge \vec{0} & \end{array} \right\}.$$

When parameter uncertainty (ambiguity) exists, the search for the min-max regret robust solution comprises finding binary choice decisions, $\vec{x}$, such that the function $\max\limits_{\omega \in \bar{\Omega}} (O_\omega^* - Z_\omega^*(\vec{x}))$ is minimized where

$$Z_\omega^*(\vec{x}) = \left\{ \begin{array}{l} \max\limits_{\vec{y}_\omega \ge \vec{0}} \quad \vec{q}_\omega{}^T \vec{y}_\omega \\ s.t. \quad W_{1\omega}\vec{y}_\omega \le \vec{h}_{1\omega} + T_{1\omega}\vec{x} \\ \quad\quad W_{2\omega}\vec{y}_\omega = \vec{h}_{2\omega} + T_{2\omega}\vec{x} \end{array} \right\} + \vec{c}_\omega{}^T \vec{x} \quad \forall \omega \in \bar{\Omega} \quad .$$

In the case when the scenario set $\bar{\Omega}$ is a finite set, the optimal choice of decision variables $\vec{x}$ (min-max regret robust solution) can be obtained by solving the following model (2):

$$\begin{array}{l} \min\limits_{\delta, \vec{x}, \vec{y}_\omega} \quad \delta \\ s.t. \left. \begin{array}{l} \delta \ge O_\omega^* - \vec{q}_\omega{}^T \vec{y}_\omega - \vec{c}_\omega{}^T \vec{x} \\ W_{1\omega}\vec{y}_\omega - T_{1\omega}\vec{x} \le \vec{h}_{1\omega} \\ W_{2\omega}\vec{y}_\omega - T_{2\omega}\vec{x} = \vec{h}_{2\omega} \\ \vec{y}_\omega \ge \vec{0} \end{array} \right\} \quad \forall \omega \in \bar{\Omega} \quad . \\ \quad\quad \vec{x} \in \{0,1\}^{|\vec{x}|} \end{array}$$

This model (2) is usually referred to as the extensive form model of the problem. If an optimal solution for the model (2) exists, the resulting binary solution is the optimal setting of decision variables $\bar{x}$. Because the model (2) has the dual structure of the block angular structure, the method of Benders (1962) is applicable and appropriate to solving these types of model when a large number of scenarios are considered. Unfortunately, when the scenario set $\bar{\Omega}$ is infinite, the problem cannot be solved directly by using the model (2).

Because of the failure of the extensive form model and the Benders' decomposition algorithm for solving a large scale problem of this type, a new algorithm, which can effectively overcome these limitations of the extensive form model and the Benders' decomposition algorithm, is proposed by Assavapokee (2004, 2008b). A key insight of this new algorithm is to recognize that, even for an infinite set of scenarios, it is possible to identify a finite set of scenarios that need to be considered as part of the iteration scheme with the use of bi-level programming. We now discuss the three-stage algorithm developed by (Assavapokee, 2004, 2008b) for handling uncertainty in model's parameters when there are infinite numbers of possible scenarios in the following section.

3.3 Three-Stage Algorithm for Interval Data Uncertainty

In the three-stage algorithm developed by (Assavapokee, 2004) to handle model parameter uncertainty they assume that each uncertain parameter can take its value from a real compact interval (infinite number of possible scenarios). The parameters in the model (1) can be classified into eight major types as shown in the following model (3):

$$\begin{aligned} \max_{\vec{x}_\omega,\vec{y}_\omega} & \ \vec{c}_\omega{}^T\vec{x}_\omega + \vec{q}_\omega{}^T\vec{y}_\omega \\ s.t. \quad & W_{1\omega}\vec{y}_\omega - T_{1\omega}\vec{x}_\omega \le \vec{h}_{1\omega} \\ & W_{2\omega}\vec{y}_\omega - T_{2\omega}\vec{x}_\omega = \vec{h}_{2\omega} \\ & \vec{x}_\omega \in \{0,1\}^{|\vec{x}_\omega|} \text{ and } \vec{y}_\omega \ge \vec{0} \end{aligned} .$$

Let the random vector $\xi = (\vec{c},\vec{q},\vec{h}_1,\vec{h}_2,T_1,T_2,W_1,W_2)$ denote the parameters defining the objective function and the constraints of the optimization problem. In this research, they assume that each component of ξ of type $\vec{c},\vec{h}_1,\vec{h}_2,T_1,$ and T_2 can independently take its values from a compact interval of real values. In other words, for the element p of the vector ξ, p can take any value from the real compact interval $[p^L, p^U]$ where p^L and p^U represent the lower and the upper bound values of the parameter p respectively. In addition, the component of ξ of type $\vec{q}$ can independently take its value from a finite set of real values. The scenario set $\bar{\Omega}$ is generated by all possible values of the parameter vector ξ. The three-stage optimization algorithm presented in (Assavapokee, 2004, 2008b) for solving min-max regret robust optimization problem under scenario set $\bar{\Omega}$ is summarized below.

3.3.1 Summary of Three-Stage Algorithm

Step 0) (Initialization) Choose a subset $\Omega \subseteq \bar{\Omega}$ and set $\Delta^U = \infty$ and $\Delta^L = 0$. Determine the value of ε (predetermined small nonnegative real value) and proceed to Step 1.

Step 1) Solve the model (1) to obtain $O^*_\omega \ \forall \omega \in \Omega$. If the model (1) is infeasible for any scenario in the scenario set Ω, the algorithm is terminated; the problem is ill-posed. Otherwise the optimal objective function value to the model (1) for scenario ω is designated as O^*_ω . Proceed to Step 2.

Step 2) (Solving the Relaxation Problem and Optimality Check) Solve the relaxation of the model (2) by considering only the scenario set Ω instead of $\bar{\Omega}$. If the relaxed model (2) is infeasible, the algorithm is terminated with the confirmation that no robust solution exists for the problem. Otherwise, set $X_\Omega = \bar{x}^*$ (optimal solution from the relaxed model (2)) and set $\Delta^L = \delta^*$ (optimal objective function value from the relaxed model (2)). If $\{\Delta^U - \Delta^L\} \leq \varepsilon$, the robust solution associated with Δ^U is the globally ε-optimal robust solution and the algorithm is terminated. Otherwise the algorithm proceeds to Step 3.

Step 3) (Feasibility Check) Solve the Bi-level-1 model specified below in section 3.5.3 by using the X_Ω information from Step 2. If the optimal objective function value of the Bi-level-1 model is nonnegative (feasible case), proceed to Step 4. Otherwise (infeasible case), $\Omega \leftarrow \Omega \cup \{\omega_1^*\}$ where ω_1^* is the infeasible scenario for X_Ω generated by the Bi-level-1 model in this iteration and return to Step 1.

Step 4) (Generate the Scenario with Maximum Regret Value for X_Ω and Optimality Check) Solve the bi-level-2 model specified below in section 3.5.2 by using the X_Ω information from Step 2. Let ω_2^* and Δ^{U*} denote the scenario with maximum regret value for X_Ω and the optimal objective function value generated by the Bi-level-2 model respectively in this iteration. Set $\Delta^U \leftarrow \min\{\Delta^{U*}, \Delta^U\}$. If $\{\Delta^U - \Delta^L\} \leq \varepsilon$, the robust solution associated with Δ^U is the globally ε-optimal robust solution and the algorithm is terminated. Otherwise, $\Omega \leftarrow \Omega \cup \{\omega_2^*\}$ and return to Step 1.

They define the algorithm Steps 1 and 2 as the first stage of the algorithm and the algorithm Step 3 and Step 4 as the second and the third stage of the algorithm respectively. Figure 3.1 illustrates the schematic structure of this algorithm.

The main purpose of the first stage is to generate a candidate robust decision, X_Ω, based on a currently considered finite small subset of scenarios, Ω and to perform the optimality check of the overall algorithm. After the candidate robust decision has been generated using the model (1) and the relaxed model (2), the optimality condition is checked. If the optimality condition is satisfied, the algorithm terminates with the globally optimal (or the ε-optimal) solution to the problem. Otherwise, the candidate robust decision, X_Ω, (the optimal setting of vector $\vec{x}$ from the relaxed model(2)) and the lower bound on the min-max regret value (the optimal objective function value of the model (2)) are then passed on to the second stage. Let Δ^L denote this lower bound value.

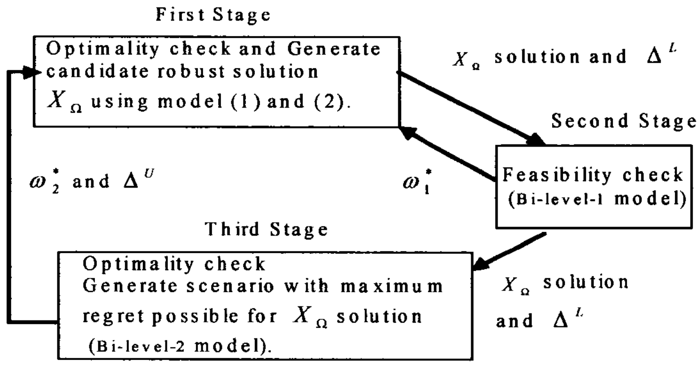


Figure 3.1: Schematic Structure of the Algorithm.

The main purpose of the second stage is to perform a feasibility check on the candidate robust decision, X_Ω forwarded from the first stage over all possible scenarios. To achieve this goal, the algorithm solves the bi-level programming model, referred to as the bi-level-1 model which is discussed extensively in section 3.5.3 If the optimal objective function value of this bi-level-1 model is negative, the algorithm has found an infeasible scenario (ω_1^*) for the candidate robust decision, X_Ω. The information of this infeasible scenario (ω^*_1) is then passed back to the first stage requesting a new candidate robust decision. Otherwise, the candidate robust decision X_Ω is feasible for all possible scenarios and the information on X_Ω and Δ^L is forwarded to the third stage of the algorithm.

The main purpose of the third stage is to determine a new scenario with the highest regret value for the current candidate robust decision, X_Ω and to perform the optimality check of the overall algorithm. To achieve these goals, the algorithm solves another bi-level programming model, referred to as the bi-level-2 model which is discussed extensively in section 3.5.2, to obtain a new scenario (ω^*_2) with the maximum regret value for X_Ω and the upper bound value on the min-max regret value (the minimum between the optimal objective function value of the Bi-level-2 model and the current upper bound value). Let Δ^U denote this upper bound value. By using the information on Δ^U and Δ^L, the algorithm will either confirm the globally optimal (or the ε-optimal) solution to the problem (if $\{\Delta^U - \Delta^L\} \leq \varepsilon$) or forward the information on scenario ω^*_2 and Δ^U to the first stage requesting a new candidate robust decision (if $\{\Delta^U - \Delta^L\} > \varepsilon$).

3.3.2 Drawbacks of 3 stage algorithm

We now discuss the existing drawbacks of this three-stage algorithm. In the first stage, the algorithm generates a candidate robust decision X_Ω by considering only the subset of scenarios Ω (relaxation problem). Though the algorithm intends to keep the size of the relaxation problem of the model (2) at the reasonable level, the number of scenarios considered by the relaxation problem may become relatively large for some practical decision problems. When the number of scenarios in the scenario set Ω

becomes large, solving the relaxed model (2) directly is definitely computationally intensive. This is a first main drawback of the existing three-stage algorithm. For this reason, the application of accelerated Benders decomposition method is proposed to improve the computational efficiency of this stage in the proposed four-stage algorithm.

In the third stage, the scenario with maximal regret possible under the current candidate robust decision is generated by using the bi-level-2 model. This bi-level-2 model has the bi-level program structure with mixed integer nonlinear structure for the leader problem and linear structure for the follower problem. The structure of the bi-level-2 model is explained in detail in Assavapokee (2004, 2008b) and in Section 3.5.2. The three-stage algorithm requires solving one bi-level-2 model in every iteration. Though the model transformation procedure that can transform the required bi-level mixed integer nonlinear optimization structure into a single-level mixed integer linear programming structure with complementary slackness constraints was presented, the required computation time for solving this mathematical model is still relatively expensive compared to other components in the overall optimization algorithm. The obvious drawback of the current algorithm in the third stage is that all candidate solutions are treated equally and they all require the execution of the bi-level-2 model. In the current algorithm, there is no logical mechanism to classify the quality of the candidate solutions. In order to improve the performance of the current algorithm, it seems logical to only seriously consider candidate solutions with promising potential and consider lightly or discard candidate solution with low or zero potential. In the proposed algorithm, we incorporate the approximation algorithms for approximately solving the bi-level-2 model in order to quantify the potential of each candidate solution. In the following sections, we propose the contributions and the procedures of the improved four-stage algorithm for the deviation robust optimization problem in detail.

3.4 Contributions of the Dissertation

The main contributions of the dissertation are

1. The proposed algorithm utilizes accelerated Benders' decomposition methodologies to speed up the solution procedure of the relaxed model (2) and the overall robust optimization algorithm. This improvement involves the use of special feasibility and optimality cuts developed especially for the problem.
2. The proposed algorithm utilizes the priority based solution procedure to quantify the quality of each candidate solution proposed from the first and second stages to minimize the number of times required in optimizing the bi-level-2 model. The algorithm requires optimizing the bi-level-2 model only for the solutions with promising potential. This framework can be achieved by utilizing appropriate approximation algorithms to approximately solve the bi-level-2 model in order to achieve the potential measurement for each candidate solution.

3.5 Four Stage Algorithm

In this section, an improved robust optimization algorithm of the algorithm developed by Assavapokee (2004, 2008b) for two-stage decision making under uncertainty and ambiguity under deviation robust criterion is presented. This improved algorithm is also an iterative algorithm. The algorithm solves the overall robust optimization problem iteratively in four computational stages. The key concept of this four-stage algorithm is also based on convergence of the upper and the lower bounds on the min-max regret value similar to the three stage algorithm developed by Assavapokee (2004, 2008b).

The four-stage algorithm utilizes the insight that solving the bi-level-2 model is computationally intensive compared to the other stages of the algorithm. Thus if we can reduce the number of times the bi-level-2 model is solved, the overall performance of the algorithm will be improved. The three-stage algorithm required solving the bi-level-2 model in every iteration. It treats each candidate solution equally and there is no method to classify the potential of each candidate solution in every iteration. In the four-stage algorithm we solve the bi-level-2 model only for those candidate solutions with promising potential by using the following screening criterion. For a given solution, $\bar{x}$, we can approximately determine its potential by calculating the quantity $\hat{\Delta}^U_{\bar{x}}$ as follow instead of directly solving the bi-level 2 model

$$\hat{\Delta}^U_{\bar{x}} = \max_{\omega \in \Omega'}\{O^*_\omega - Z^*_\omega(\bar{x})\} \text{ where } \Omega' \subseteq \bar{\Omega} \quad .$$

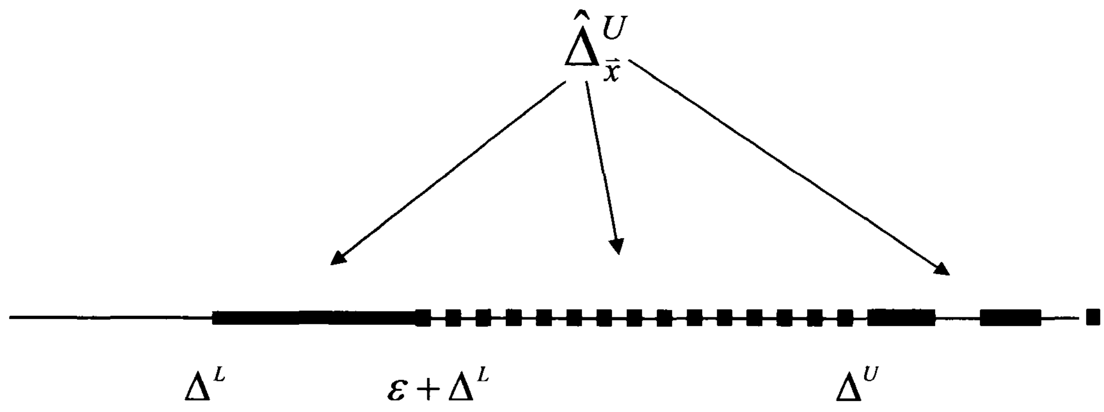
Note that $\hat{\Delta}^U_{\bar{x}} \leq \max_{\omega \in \bar{\Omega}}\{O^*_\omega - Z^*_\omega(\bar{x})\} = \Delta^U_{\bar{x}}$ (objective function from the bi-level-2 model)

Let us define the notation $\omega^*_{\bar{x}} \in \arg\max_{\omega \in \bar{\Omega}}\{O^*_\omega - Z^*_\omega(\bar{x})\}$.

Given the value of $\hat{\Delta}^U_{\bar{x}}$, we will use the following rules to classify the quality of the solution $\bar{x}$:

1. A solution $\bar{x}$ is classified as the solution with no potential if $\hat{\Delta}^U_{\bar{x}} \geq \Delta^U$. This type of solution will not be considered further by the algorithm.
2. A solution $\bar{x}$ is classified as a solution with possible potential if $\Delta^U > \hat{\Delta}^U_{\bar{x}} > \varepsilon + \Delta^L$. This type of solution will be recorded into the set of possible solutions for further consideration by the algorithm.
3. A solution $\bar{x}$ is classified as a solution with promising potential if $\varepsilon + \Delta^L \geq \hat{\Delta}^U_{\bar{x}}$. The bi-level-2 model will be solved for the solution of this type to calculate the value of $\Delta^U_{\bar{x}}$ in order to update the value of Δ^U.

Figure 3.2 illustrates the classification procedure.



$$\hat{\Delta}_{\bar{x}}^{U} = \max_{\omega \in \Omega'} \{O_\omega^* - Z_\omega^*(\bar{x})\} \text{ where } \Omega' \subseteq \bar{\Omega}$$

Figure 3.2: Classification of candidate solutions

As described below, we propose a four-stage optimization algorithm for solving min-max regret robust optimization problem under scenario set $\overline{\Omega}$ that proceeds as follows. Figure 3.2 illustrates the schematic structure of this new algorithm.

<u>Summary of Four-Stage Algorithm</u>

Step 0) (Initialization) Choose a finite subset $\Omega \subseteq \bar{\Omega}$ and set $\Delta^U = \infty$ and $\Delta^L = 0$. Determine the value of ε (predetermined small nonnegative real value), set $S = \phi$, and proceed to Step 1.

Step 1) Solve the model (1) to obtain O_ω^* $\forall \omega \in \Omega$. If the model (1) is infeasible for any scenario in the scenario set Ω, the algorithm is terminated; the problem is ill-posed. Otherwise the optimal objective function value to the model (1) for scenario ω is designated as O_ω^* . Proceed to Step 2.

Step 2) (Solving the Relaxation Problem and Optimality Check) Solve the relaxation of the model (2) by considering only the scenario set Ω instead of $\bar{\Omega}$ by using the

Benders' decomposition algorithm. If the relaxed model (2) is infeasible, the algorithm is terminated with the confirmation that no robust solution exists for the problem. Otherwise, set $X_\Omega = \vec{x}^*$ (optimal solution from the relaxed model (2)) and set $\Delta^L = \delta^*$ (optimal objective function value from the relaxed model (2)). If $\{\Delta^U - \Delta^L\} \le \varepsilon$, the robust solution associated with Δ^U is the globally ε-optimal robust solution and the algorithm is terminated. Otherwise the algorithm proceeds to Step 3.

Step 3) (Prioritizing candidate solutions and Optimality check) Check the list of potential solutions S for the solution X' such that $\hat{\Delta}^U_{X'} - \Delta^L \le \varepsilon$.

If such a solution exists, $S \leftarrow S - \{X'\}$ and the bi-level-2 model is solved under the fixed setting of the discrete solution X', Otherwise the algorithm proceeds to Step 4 along with the current candidate robust solution (X_Ω) and the updated Δ^U and Δ^L.

The results from this bi-level-2 model are the maximum regret value $\Delta^U_{X'}$ associated with the solution X' and a scenario $\omega^*_{X'}$ associated with this maximum regret value of X'.

Set $\Omega \leftarrow \Omega \cup \{\omega^*_{X'}\}$, and then set $\Delta^U \leftarrow \min\{\Delta^U_{X'}\ ,\ \Delta^U\}$.

If $\{\Delta^U - \Delta^L\} \le \varepsilon$, the robust solution associated with Δ^U is the globally ε-optimal robust solution and the algorithm is terminated. Otherwise set $S \leftarrow S - D$ such that $D = \{X \in S \mid \hat{\Delta}^U_X \ge \Delta^U\}$. The algorithm then repeats the Step 3.

Step 4) (Feasibility Check) Solve the bi-level-1 model by using the X_Ω information from Step 3. If the optimal objective function value of the bi-level-1 model is nonnegative (feasible case), proceed to Step 5. Otherwise (infeasible case), $\Omega \leftarrow \Omega \cup \{\omega^*_1\}$ where ω^*_1 is the infeasible scenario for X_Ω generated by the bi-level-1 model in this iteration and return to Step 1.

Step 5) (Prioritizing candidate solutions) Solve the bi-level-3 model by using the X_Ω information to determine a new scenario ω' with the highest approximated regret value $\hat{\Delta}^U_{X\Omega}$ for the current candidate robust solution (X_Ω). If $\hat{\Delta}^U_{X\Omega} \geq \Delta^U$, set $\Omega \leftarrow \Omega \cup \{\omega'\}$ and the algorithm proceeds to Step 1. Otherwise the algorithm proceeds to Step 6.

Step 6) Set $S \leftarrow S \cup \{X_\Omega\}$. If $\hat{\Delta}^U_{X_\Omega} - \Delta^L \leq \varepsilon$, proceed to Step 7.

Otherwise return to Step1.

Step 7) set $S \leftarrow S - \{X_\Omega\}$ and solve the bi-level-2 model for the discrete solution X_Ω. Let $\omega^*_{X_\Omega}$ and $\Delta^U_{X_\Omega}$ denote the scenario with maximum regret value for X_Ω and the optimal objective function value generated by the bi-level-2 model respectively in this iteration. The new generated scenario $\omega^*_{X_\Omega}$ is then added to the set Ω. Set $\Delta^U \leftarrow$ min $\{\Delta^U_{X_\Omega}, \Delta^U\}$. If $\{\Delta^U - \Delta^L\} \leq \varepsilon$, the robust solution associated with Δ^U is the globally ε-optimal robust solution and the algorithm is terminated. Otherwise, set $S \leftarrow S - D$ such that $D = \{X \in S \mid \hat{\Delta}^U_X \geq \Delta^U\}$ and the algorithm returns to Step 1.

The Steps 1 and 2 are defined as the first stage of the algorithm, Step 3 as the second stage, Step 4 as the third stage and Steps 5, 6, and 7 as the fourth stage of the algorithm respectively. Figure 3.3 illustrates a schematic structure of this algorithm.

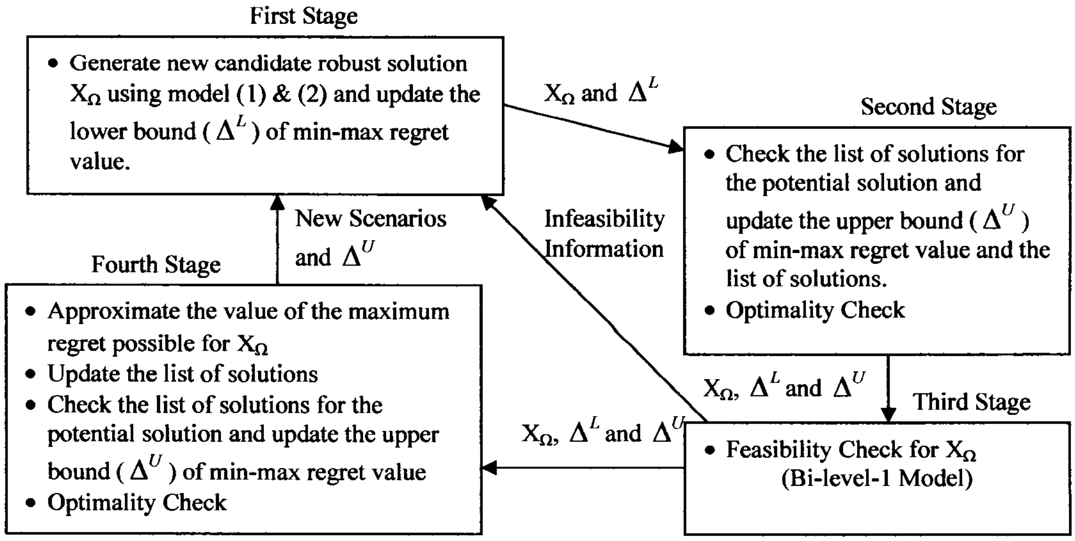


Figure 3.3: Schematic Structure of Proposed Four-Stage Algorithm

Similar to the first stage of the algorithm presented in Figure 3.1, the first stage of this new algorithm starts by initializing required parameter values and generating a candidate robust decision, X_Ω, and the lower bound on min-max regret value (Δ^L) based on a considered finite subset of scenarios Ω by using the model (1) and the relaxed model (2). This information is then passed on to the second stage.

The second stage starts by checking if there is any solution with promising potential in the current potential solution list *S*. The promising potential solution, *X'*, is the solution such that $\hat{\Delta}^U_{X'} - \Delta^L \le \varepsilon$. If the promising potential solution exists, it is removed from *S* and the bi-level-2 model is optimally solved for this solution. The results from solving the bi-level-2 model are a new generated scenario and the maximum regret value ($\Delta^U_{X'}$) for the selected potential solution. This new scenario is then added to the scenario set Ω and we update $\Delta^U \leftarrow \min\{\Delta^U_{X'}, \Delta^U\}$. By using the Δ^U and Δ^L information, the optimality condition can be checked ($\Delta^U - \Delta^L \le \varepsilon$). If the optimality condition is satisfied, the algorithm terminates with the solution associated with the maximum regret value of Δ^U as a ε-optimal robust solution. Otherwise, the algorithm update the potential solution list by deleting any solution, X, in the list such that $\hat{\Delta}_X^U \geq \Delta^U$. This process is repeated until there is no more promising potential solution in S. If this is the case, the candidate robust decision (X_Ω) and the updated Δ^U and Δ^L information are forwarded to the third stage of the algorithm.

Similar to the second stage of the algorithm presented in Figure 3.1, the main purpose of the third stage is to perform a feasibility check on the candidate robust decision over all possible scenarios by using the bi-level-1 model. After the bi-level-1 model is transformed and solved, if there is any infeasible scenario under the current candidate robust decision, the information of the infeasible scenario is detected and is sent back to the first stage requesting a new candidate robust decision. Otherwise, the candidate robust decision (X_Ω) and the updated upper and lower bound information are forwarded to the fourth stage of the algorithm.

The main purpose of the fourth stage is to determine a new scenario with the highest approximated regret value ($\hat{\Delta}_{X_\Omega}^U$) for the current candidate robust decision (X_Ω) by using approximation algorithms. Note that the approximation is performed in such a way that $\hat{\Delta}_{X_\Omega}^U \leq \Delta_{X_\Omega}^U$. After this approximation is performed, if $\hat{\Delta}_{X_\Omega}^U \geq \Delta^U$, the generated scenario is added to the scenario set Ω and the first stage is repeated. Otherwise, the candidate robust solution X_Ω is added to S. The algorithm then checks whether the candidate robust solution is a promising potential solution. If this is the case, it is removed from S and the bi-level-2 model is solved for this solution. The new generated scenario is then added to the scenario set Ω and the Δ^U value is updated, $\Delta^U \leftarrow \min\{\Delta^U_{X_\Omega}\ ,\ \Delta^U\}$. By using the Δ^U and Δ^L information, the optimality condition can be checked ($\Delta^U - \Delta^L \le \varepsilon$). If the optimality condition is satisfied, the algorithm terminates with the solution associated with the maximum regret value of Δ^U as a ε-optimal robust solution. Otherwise, the algorithm updates S by deleting any solution, X, such that $\hat{\Delta}^U_X \ge \Delta^U$ and the first stage is repeated. In following sections, we discuss the methodologies utilized in each stage of the algorithm in detail.

3.5.1 The First Stage

The purposes of the first stage are (1) to find $\vec{x}_\Omega \in \arg\min_{\vec{x}}\left\{\max_{\omega\in\Omega}\{O^*_\omega - Z^*_\omega(\vec{x})\}\right\}$, (2) to find $\Delta^L = \min_{\vec{x}}\max_{\omega\in\Omega}\{O^*_\omega - Z^*_\omega(\vec{x})\}$, and (3) to determine if the algorithm has discovered an optimal robust solution for the problem. The first stage utilizes two main optimization models: the model (1) and the relaxed model (2). The model (1) is used to calculate O^*_ω for all scenarios $\omega \in \Omega \subseteq \bar{\Omega}$. If the model (1) is infeasible for any scenario $\omega \in \Omega$, the algorithm is terminated with the conclusion that there exists no robust solution to the problem. Otherwise, once all required values of $O^*_\omega\ \forall\omega\in\Omega$ are obtained, the relaxed model (2) is solved. In this research, we recommend solving the relaxed model (2) by utilizing the Benders' decomposition techniques as follow. The relaxed model (2) has the structure represented by model (4):

$$
\begin{aligned}
&\min_{\vec{x},\vec{y}_\omega}\left(\max_{\omega\in\Omega}(O^*_\omega - \vec{c}^{\,T}_\omega\vec{x} - \vec{q}^{\,T}_\omega\vec{y}_\omega)\right)\\
&\text{s.t.}\ \left.\begin{aligned}
&\boldsymbol{W}_{1\omega}\vec{y}_\omega - \boldsymbol{T}_{1\omega}\vec{x} \le \vec{h}_{1\omega}\\
&\boldsymbol{W}_{2\omega}\vec{y}_\omega - \boldsymbol{T}_{2\omega}\vec{x} = \vec{h}_{2\omega}\\
&\vec{y}_\omega \ge \vec{0}
\end{aligned}\right\}\ \forall\omega\in\Omega\\
&\qquad \vec{x}\in\{0,1\}^{|\vec{x}|}
\end{aligned}
$$

This model can also be rewritten as the following model 5:

$$\min\{f(\vec{x}) \mid \vec{x} \in \{0,1\}^{|\vec{x}|}\} \text{ where } f(\vec{x}) = \max_{\omega\in\Omega}(O_\omega^* - Q_\omega(\vec{x}) - c_\omega^T\vec{x}) \text{ and}$$

$$\begin{aligned} Q_\omega(\vec{x}) = \max_{\vec{y}_\omega \geq \vec{0}} \quad & \vec{q}_\omega^T \vec{y}_\omega \\ \text{s.t.} \quad & \boldsymbol{W}_{1\omega}\vec{y}_\omega \leq \vec{h}_{1\omega} + \boldsymbol{T}_{1\omega}\vec{x} \qquad (\vec{\pi}_{1,\omega,\vec{x}}) \\ & \boldsymbol{W}_{2\omega}\vec{y}_\omega = \vec{h}_{2\omega} + \boldsymbol{T}_{2\omega}\vec{x} \qquad (\vec{\pi}_{2,\omega,\vec{x}}) \end{aligned}$$

where the symbols in parenthesis next to the constraints denote to the corresponding dual variables. The results from the following two lemmas are used to generate the master problem and sub problems of the Benders' decomposition for the relaxed model (2).

***Lemma* 1:** $f(\vec{x})$ is a convex function on $\vec{x}$.

Proof: $f(\vec{x}) = \max_{\omega\in\Omega}(O_\omega^* - Q_\omega(\vec{x}) - c_\omega^T\vec{x})$ is a convex function on $\vec{x}$ because of the following reasons. (1) $Q_\omega(\vec{x})$ and $c_\omega^T\vec{x}$ are concave functions on $\vec{x}$;

(2) (–1)*concave function is a convex function; (3) Summation of convex functions is also a convex function; and (4) Maximum function of convex functions is also a convex function.

***Lemma* 2:** $\left(-\vec{c}_{\omega(i)}^T - (\vec{\pi}_{1,\omega(i),\vec{x}(i)}^*)^T \boldsymbol{T}_{1\omega(i)} - (\vec{\pi}_{2,\omega(i),\vec{x}(i)}^*)^T \boldsymbol{T}_{2\omega(i)}\right)^T \in \partial f(\vec{x}(i))$ where $\omega(i) \in \arg\max_{\omega\in\Omega}\{O_\omega^* - \vec{c}_\omega^T\vec{x}(i) - Q_\omega(\vec{x}(i))\}$, $\partial f(\vec{x}(i))$ is sub-differential of the function f at $\vec{x}(i)$ and $(\vec{\pi}_{1,\omega(i),\vec{x}(i)}^*, \vec{\pi}_{2,\omega(i),\vec{x}(i)}^*)$ is the optimal solution of the dual problem in the calculation of $Q_\omega(\vec{x})$ when $\omega = \omega(i)$ and $\vec{x} = \vec{x}(i)$.

Proof: From duality theory:

$$Q_{\omega(i)}(\vec{x}(i)) = (\vec{\pi}_{1,\omega(i),\vec{x}(i)}^*)^T(\vec{h}_{1\omega(i)} + \boldsymbol{T}_{1\omega(i)}\vec{x}(i)) + (\vec{\pi}_{2,\omega(i),\vec{x}(i)}^*)^T(\vec{h}_{2\omega(i)} + \boldsymbol{T}_{2\omega(i)}\vec{x}(i))$$

and $Q_{\omega(i)}(\vec{x}) \leq (\vec{\pi}_{1,\omega(i),\vec{x}(i)}^*)^T(\vec{h}_{1\omega(i)} + \boldsymbol{T}_{1\omega(i)}\vec{x}) + (\vec{\pi}_{2,\omega(i),\vec{x}(i)}^*)^T(\vec{h}_{2\omega(i)} + \boldsymbol{T}_{2\omega(i)}\vec{x})$ for arbitrary $\vec{x}$.

Thus, $Q_{\omega(i)}(\vec{x}) - Q_{\omega(i)}(\vec{x}(i)) \leq (\vec{\pi}_{1,\omega(i),\vec{x}(i)}^*)^T \boldsymbol{T}_{1\omega(i)}(\vec{x} - \vec{x}(i)) + (\vec{\pi}_{2,\omega(i),\vec{x}(i)}^*)^T \boldsymbol{T}_{2\omega(i)}(\vec{x} - \vec{x}(i))$ and

$$O^*_{\omega(i)} - Q_{\omega(i)}(\vec{x}) \geq O^*_{\omega(i)} - Q_{\omega(i)}(\vec{x}(i)) - \left((\vec{\pi}^*_{1,\omega(i),\vec{x}(i)})^T \boldsymbol{T}_{1\omega(i)} + (\vec{\pi}^*_{2,\omega(i),\vec{x}(i)})^T \boldsymbol{T}_{2\omega(i)}\right)(\vec{x} - \vec{x}(i)).$$

From $-\vec{c}^T_{\omega(i)}\vec{x} = -\vec{c}^T_{\omega(i)}\vec{x}(i) - \vec{c}^T_{\omega(i)}(\vec{x} - \vec{x}(i))$ and $f(\vec{x}) = \max_{\omega \in \Omega}(O^*_\omega - Q_\omega(\vec{x}) - c^T_\omega \vec{x})$,

$$\begin{aligned} f(\vec{x}) &\geq O^*_{\omega(i)} - Q_{\omega(i)}(\vec{x}) - \vec{c}^T_{\omega(i)}\vec{x} \\ &\geq O^*_{\omega(i)} - Q_{\omega(i)}(\vec{x}(i)) - \vec{c}^T_{\omega(i)}\vec{x}(i) + \left(-\vec{c}^T_{\omega(i)} - (\vec{\pi}^*_{1,\omega(i),\vec{x}(i)})^T \boldsymbol{T}_{1\omega(i)} - (\vec{\pi}^*_{2,\omega(i),\vec{x}(i)})^T \boldsymbol{T}_{2\omega(i)}\right)(\vec{x} - \vec{x}(i)) \end{aligned}.$$

From $\omega(i) \in \arg\max_{\omega \in \Omega}\{O^*_\omega - \vec{c}^T_\omega \vec{x}(i) - Q_\omega(\vec{x}(i))\}$,

$$f(\vec{x}) \geq f(\vec{x}(i)) + \left(-\vec{c}^T_{\omega(i)} - (\vec{\pi}^*_{1,\omega(i),\vec{x}(i)})^T \boldsymbol{T}_{1\omega(i)} - (\vec{\pi}^*_{2,\omega(i),\vec{x}(i)})^T \boldsymbol{T}_{2\omega(i)}\right)(\vec{x} - \vec{x}(i)) .$$

Based on the results of the Lemma 1 and 2, we briefly state the general Benders decomposition algorithm as it applies to the relaxed model (2).

Benders Decomposition Algorithm for the Relaxed Model (2):

***Step* 0:** Set lower and upper bounds $lb = -\infty$ and $ub = +\infty$ respectively. Set the iteration counter $k = 0$. Let Y^0 includes all cuts generated from all previous iterations of the proposed three-stage algorithm. All these cuts are valid because the proposed algorithm always add more scenarios to the set Ω and this causes the feasible region of the relaxed model (2) to shrink from one iteration to the next. Let $\vec{x}^*$ denote the incumbent solution.

***Step* 1:** Solve the master problem

$$\begin{aligned} lb = \min_{\theta,\vec{x}} \ & \theta \\ \text{s.t. } & \theta \geq \vec{a}^T_i \vec{x} + b_i \qquad \forall i = 1,2,\dots,k \\ & (\theta, \vec{x}) \in Y^k \end{aligned} \qquad (3.5.1.1)$$

If the master problem is infeasible, stop and report that the relaxed model (2) is infeasible. Otherwise, update $k = k + 1$ and let $\vec{x}(k)$ be an optimal solution of the master problem.

***Step* 2:** For each $\omega \in \Omega$, solve the following sub problem

$$
\begin{aligned}
Q_\omega(\vec{x}(k)) = \max_{\vec{y}_\omega \geq \vec{0}} \ & \vec{q}_\omega^T \vec{y}_\omega \\
\text{s.t.} \ & \boldsymbol{W}_{1\omega} \vec{y}_\omega \leq \vec{h}_{1\omega} + \boldsymbol{T}_{1\omega} \vec{x}(k) \qquad (\vec{\pi}_{1,\omega,\vec{x}(k)}) \\
& \boldsymbol{W}_{2\omega} \vec{y}_\omega = \vec{h}_{2\omega} + \boldsymbol{T}_{2\omega} \vec{x}(k) \qquad (\vec{\pi}_{2,\omega,\vec{x}(k)})
\end{aligned} \tag{3.5.1.2}
$$

where the symbols in parenthesis next to the constraints denote to the corresponding dual variables. If the sub problem is infeasible for any scenario $\omega \in \Omega$, go to Step 5. Otherwise, using the sub problem objective values, compute the objective function value $f(\vec{x}(k)) = O^*_{\omega(k)} - \vec{c}^T_{\omega(k)} \vec{x}(k) - Q_{\omega(k)}(\vec{x}(k))$ corresponding to the current feasible solution $\vec{x}(k)$ where $\omega(k) \in \arg\max_{\omega \in \Omega} \{O^*_\omega - \vec{c}^T_\omega \vec{x}(k) - Q_\omega(\vec{x}(k))\}$. If $ub > f(\vec{x}(k))$, update the upper bound $ub = f(\vec{x}(k))$ and the incumbent solution $\vec{x}^* = \vec{x}(k)$.

***Step* 3:** If $ub - lb \leq \lambda$, where $\lambda \geq 0$ is a pre-specified tolerance, stop and return $\vec{x}^*$ as the optimal solution and ub as the optimal objective value; otherwise proceed to Step 4.

***Step* 4:** For the scenario $\omega(k) \in \arg\max_{\omega \in \Omega} \{O^*_\omega - \vec{c}^T_\omega \vec{x}(k) - Q_\omega(\vec{x}(k))\}$, let $(\vec{\pi}^*_{1,\omega(k),\vec{x}(k)}, \vec{\pi}^*_{2,\omega(k),\vec{x}(k)})$ be the optimal dual solutions for the sub problem corresponding to $\vec{x}(k)$ and $\omega(k)$ solved in Step 2. Compute the cut coefficients $\vec{a}^T_k = -\left(\vec{c}^T_{\omega(k)} + (\vec{\pi}^*_{1,\omega(k),\vec{x}(k)})^T \boldsymbol{T}_{1\omega(k)} + (\vec{\pi}^*_{2,\omega(k),\vec{x}(k)})^T \boldsymbol{T}_{2\omega(k)}\right)$, and $b_k = -\vec{a}^T_k \vec{x}(k) + f(\vec{x}(k))$, and go to Step 1.

***Step* 5:** Let $\hat{\omega} \in \Omega$ be a scenario such that the sub problem is infeasible. Solve the following optimization problem where $\vec{0}$ and $\vec{1}$ represent the vector with all elements equal to zero and one respectively.

$$\min_{\vec{v}_1,\vec{v}_2}(\vec{h}_{1\hat{\omega}} + T_{1\hat{\omega}}\vec{x}(k))^T\vec{v}_1 + (\vec{h}_{2\hat{\omega}} + T_{2\hat{\omega}}\vec{x}(k))^T\vec{v}_2$$
$$\text{s.t.}\quad W_{1\hat{\omega}}^T\vec{v}_1 + W_{2\hat{\omega}}^T\vec{v}_2 \geq \vec{0} \qquad (3.5.1.3)$$
$$\vec{0} \leq \vec{v}_1 \leq \vec{1}, \quad -\vec{1} \leq \vec{v}_2 \leq \vec{1}$$

Let $\vec{v}_1^*$ and $\vec{v}_2^*$ be the optimal solution of this optimization problem. Set $k = k - 1$ and $Y^k \leftarrow Y^k \cup \{\vec{x} \mid (\vec{h}_{1\hat{\omega}} + T_{1\hat{\omega}}\vec{x})^T\vec{v}_1^* + (\vec{h}_{2\hat{\omega}} + T_{2\hat{\omega}}\vec{x})^T\vec{v}_2^* \geq 0\}$ and go to Step 1.

If the relaxed model (2) is infeasible, the algorithm is terminated with the conclusion that there exists no robust solution to the problem. Otherwise, its results are the candidate robust decision, $\vec{x}_\Omega = \vec{x}^*$, and the lower bound on min-max regret value, $\Delta^L = \max_{\omega\in\Omega}\{O_\omega^* - Z_\omega^*(\vec{x}^*)\}$ obtained from the relaxed model (2). The optimality condition is then checked. The optimality condition will be satisfied when $\Delta^U - \Delta^L \leq \varepsilon$, where $\varepsilon \geq 0$ is pre-specified tolerance. If the optimality condition is satisfied, the algorithm is terminated with the ε-optimal robust solution. Otherwise the solution $\vec{x}_\Omega$ and the value of Δ^L are forwarded to the second stage.

3.5.2 The Second Stage

In the second stage, we first check the set of solutions S for the solution with promising potential. A solution X' is said to be the solution with promising potential if and only if $\hat{\Delta}_{X'}^U - \Delta^L \leq \varepsilon$ where $\hat{\Delta}_{\vec{x}}^U = \max_{\omega\in\Omega'}\{O_\omega^* - Z_\omega^*(\vec{x})\}$ and $\Omega' \subseteq \bar{\Omega}$. The methodology for calculating the $\hat{\Delta}_{X'}^U$ value is explained in detail in Section 3.5.4. If such a solution X' exists, it is first removed from the set of solutions S and the bi-level-2 model is solved under the fixed setting of the discrete solution X'. Otherwise the algorithm proceeds to the third stage with the current candidate robust solution ($\vec{x}_\Omega$) and the updated values of Δ^U and Δ^L.

The main purpose of the bi-level-2 model is to identify a scenario $\omega^*_{X'}$ associated with the largest regret value for the solution X'. In other words, the bi-level-2 model searches for the scenario $\omega^*_{X'}$ such that

$$\omega^*_{X'} \in \arg\max_{\omega\in\Omega}\{O^*_\omega - Z^*_\omega(X')\} \quad . \tag{3.5.2.1}$$

In the bilevel-2 model, the leader problem is tasked with finding the setting of the vector $\vec{\xi} = (\vec{c}, T_1, T_2, \vec{h}, \vec{g}, \vec{q})$ and the vector $(\vec{x}_1, \vec{y}_1)$ that result in the maximum regret value overall possible scenarios, $\max_{\omega\in\Omega}\{O^*_\omega - Z^*_\omega(X')\}$, for the candidate robust solution X'. The follower problem on another hand is tasked to respond with the setting of vector $\vec{y}_2$ that maximizes the value of $Z^*_\omega(X')$, under the setting of the vector $\vec{\xi}$ established by the leader problem. The structure of the bilevel-2 model is represented by the model (5):

$$
\begin{array}{lll}
\max & q^T \vec{y}_1 + \vec{c}^T \vec{x}_1 - \vec{q}^T \vec{y}_2 - \vec{c}^T X' & \\
\text{s.t.} & W_1 \vec{y}_1 \le \vec{h}_1 + T_1 \vec{x}_1 & \\
& W_2 \vec{y}_1 = \vec{h}_2 + T_2 \vec{x}_1 & \\
& c_l^L \le c_l \le c_l^U, & \forall l \\
& T_{1il}^L \le T_{1il} \le T_{1il}^U, & \forall i \in L, \ \forall l \\
& T_{2il}^L \le T_{2il} \le T_{2il}^U, & \forall i \in E, \ \forall l \\
& h_{1i}^L \le h_{1i} \le h_{1i}^U, & \forall i \in L \\
& h_{2i}^L \le h_{2i} \le h_{2i}^U, & \forall i \in E \\
& q_j \in \{q_{j(1)}, q_{j(2)}, \dots, q_{j(mj)}\} & \forall j \\
& \vec{y}_1 \ge \vec{0} \quad \vec{x}_1 \in \{0,1\}^{|\vec{x}_1|} & \\
& \max \ \vec{q}^T \vec{y}_2 & \\
& \text{s.t.} \quad W_1 \vec{y}_2 \le \vec{h}_1 + T_1 X' & \\
& \qquad W_2 \vec{y}_2 = \vec{h}_2 + T_2 X' & \\
& \qquad \vec{y}_2 \ge \vec{0} &
\end{array}
$$

The solution methodology for solving the model (5) consists of the following two main steps.

3.5.2.1 Parameter Preprocessing Step

From the structure of the model (5), some elements of the vector $\bar{\xi}$ can be predetermined to attain their optimal setting at one of their bounds.

Preprocessing Step for c: Each element c_l of the parameter vector $\bar{c}$ is represented in the objective function of the model (5) as $c_l x_{1l} - c_l X'_l$. From any given value of X'_l, the value of c_l can be predetermined by the following simple rules. If the value of X'_l is one, the optimal setting of c_l is $c_l^* = c_l^L$. Otherwise, the optimal setting of c_l is $c_l^* = c_l^U$.

Preprocessing Step for T: Each element T_{1il} of the parameter matrix T is presented in the functional constraints of the model (5) as

$$\sum_j W_{1ij} y_{1j} \le h_{1i} + T_{1il} x_{1l} + \sum_{k \ne l} T_{1ik} x_{1k} \text{ and } \sum_j W_{1ij} y_{2j} \le h_{1i} + T_{1il} X'_l + \sum_{k \ne l} T_{1ik} X'_k .$$

For any given X' information, the value of T_{1il} can be predetermined at T_{1il}^U if the value of X'_l is zero. In the case when the value of X'_l is one, the optimal setting of T_{1il} satisfies the following set of constraints illustrated in eq. 3.5.2.1.1 where the new variable TX_{1il} replaces the nonlinear term $T_{1il} x_{1l}$ in the model (5). The insight of this set of constraints eq. 3.5.2.1.1 is that, if the value of x_{1l} is set to be zero by the model, the optimal setting of T_{1il} is T_{1il}^L. Otherwise, the optimal setting of T_{1il} can take any value from the compact interval $[T_{1il}^L, T_{1il}^U]$

$$
\begin{aligned}
&TX_{1il} - T_{1il} + T_{1il}^{L}(1 - x_{1l}) \leq 0 \\
&-TX_{1il} + T_{1il} - T_{1il}^{U}(1 - x_{1l}) \leq 0 \\
&T_{1il}^{L} x_{1l} \leq TX_{1il} \leq T_{1il}^{U} x_{1l} \\
&T_{1il} \leq T_{1il}^{L} + x_{1l}(T_{1il}^{U} - T_{1il}^{L}) \\
&T_{1il}^{L} \leq T_{1il} \leq T_{1il}^{U}
\end{aligned}
\quad . \qquad (3.5.2.1.1)
$$

<u>Preprocessing Step for q:</u> Each element q_j of the parameter vector $\vec{q}$ is presented in the objective function of the model (5) as $q_j y_{1j} - q_j y_{2j}$. Each parameter q_j can independently take its values from the ascending ordered set of real values $\{q_{j(1)}, q_{j(2)}, ..., q_{j(mj)}\}$, where mj represents the number of possible values for q_j. For simplicity, the notations q_j^L and q_j^U are used to represent the terms $q_{j(1)}$ and $q_{j(mj)}$ respectively. For any given X' information, in the case where the value of y_{2j} is forced by other parameters setting to be zero, the parameter q_j value can be predetermined to be $q_j^* = q_j^U$. In other cases, we add the decision variables QY_{1j} and QY_{2j} to replace the terms $q_j y_{1j}$ and $q_j y_{2j}$ respectively in model (5) and a set of variables and constraints illustrated in eq. 3.5.2.1.2 to replace the constraint $q_j \in \{q_{j(1)}, q_{j(2)}, ..., q_{j(mj)}\}$ in the model (5) where y_{rj}^U and y_{rj}^L represent the upper bound and the lower bound of variable y_{rj} respectively for r = 1, 2. A Special Ordered Set of type One (SOS1) is defined to be a set of variables for which not more than one member from the set may be non-zero.

$$q_j = \sum_{s=1}^{mj} q_{j(s)} bi_{j(s)}, \ \sum_{s=1}^{mj} bi_{j(s)} = 1, \ bi_{j(s)} \ge 0 \ \forall s \in \{1,2,...,mj\} \text{ and } \bigcup_{\forall s}\{bi_{j(s)}\} \text{ is SOS1}$$

$$QY_{1j} = \sum_{s=1}^{mj} q_{j(s)} z_{1j(s)}$$

$$y_{1j}^{L} bi_{j(s)} \le z_{1j(s)} \le y_{1j}^{U} bi_{j(s)}, \ z_{1j(s)} \le y_{1j} - y_{1j}^{L}(1 - bi_{j(s)}),$$

$$z_{1j(s)} \ge y_{1j} - y_{1j}^{U}(1 - bi_{j(s)}) \quad \forall s \in \{1,...,mj\}$$

$$QY_{2j} = \sum_{s=1}^{mj} q_{j(s)} z_{2j(s)} \qquad (3.5.2.1.2)$$

$$y_{2j}^{L} bi_{j(s)} \le z_{2j(s)} \le y_{2j}^{U} bi_{j(s)}, \ z_{2j(s)} \le y_{2j} - y_{2j}^{L}(1 - bi_{j(s)}),$$

$$z_{2j(s)} \ge y_{2j} - y_{2j}^{U}(1 - bi_{j(s)}) \quad \forall s \in \{1,...,mj\}$$

It is worth pointing out that the optimal setting of each parameter in the bilevel-2 model does not always reside at its upper or lower bounds.

3.5.2.2 Problem Transformation Step

Because the follower problem of the model (5) has a linear program structure and it affects the leader decisions only through its objective function, the follower problem can be replaced by the explicit representation of its optimality conditions including primal constraints, dual constraints, and complementary slackness conditions. Thus, the model (5) can be transformed into a single level mixed integer nonlinear programming problem with complementary slackness constraints as shown in model (6):

$$
\begin{aligned}
& \max\{\vec{q}^T \vec{y}_1 + \vec{c}^T \vec{x}_1 - \vec{q}^T \vec{y}_2 - \vec{c}^T X'\} \\
\text{s.t.}\quad & W_1 \vec{y}_1 \le \vec{h}_1 + T_1 \vec{x}_1 \\
& W_2 \vec{y}_1 = \vec{h}_2 + T_2 \vec{x}_1 \\
& W_1 \vec{y}_2 = \vec{h}_1 - \vec{s}_1 + T_1 X' \\
& W_2 \vec{y}_2 = \vec{h}_2 + T_2 X' \\
& W_1^T \vec{w}_1 + W_2^T \vec{w}_2 - \vec{a} = \vec{q} \\
& w_{1i} s_{1i} = 0 && \forall i \in L \\
& a_j y_{2j} = 0 && \forall j \\
& c_l^L \le c_l \le c_l^U && \forall l \\
& T_{1il}^L \le T_{1il} \le T_{1il}^U && \forall i \in L \quad \forall l \\
& T_{2il}^L \le T_{2il} \le T_{2il}^U && \forall i \in E \quad \forall l \\
& h_{1i}^L \le h_{1i} \le h_{1i}^U && \forall i \in L \\
& h_{2i}^L \le h_{2i} \le h_{2i}^U && \forall i \in E \\
& q_j \in \{q_{j(1)}, q_{j(2)}, \ldots, q_{j(mj)}\} && \forall j \\
& \vec{y}_1 \ge \vec{0}, \vec{y}_2 \ge \vec{0}, \vec{s}_1 \ge \vec{0}, \vec{a} \ge \vec{0}, \vec{w}_1 \ge \vec{0}, \vec{x}_1 \in \{0,1\}^{|\vec{x}_1|}
\end{aligned}
$$

Finally, the model (6) is transformed into a single level MILP problem with complementary slackness constraints as shown in the model (7) by including all additional constraints and variables presented in the preprocessing steps. The last step is to handle the complementary slackness conditions. The direct approach of Bard and Moore (1990) is used, in which the constraints are branched directly rather than using a classical relaxation method. The latter approach has been shown to be ineffective (Assavapokee 2004) for bi-level programming problems because high numerical precision is required to avoid the leader problem perturbing the follower problem optimal solution .

Model 7:

$$
\begin{aligned}
\max\ \Delta^{U*} = & \sum_{j|Ind\_q_j=1} QY_{1j} + \sum_{j|Ind\_q_j=0} q_j^* y_{1j} + \sum_l c_l^* x_{1l} - \sum_{j|Ind\_q_j=1} QY_{2j} \\
& - \sum_{j|Ind\_q_j=0} q_j^* y_{2j} - \sum_l c_l^* X'_l
\end{aligned}
$$

$$
\begin{aligned}
\text{s.t.}\ & \sum_j W_{1ij} y_{1j} - \sum_{l|Ind\_T_{1il}=1} TX_{1il} - \sum_{l|Ind\_T_{1il}=0} T_{1il}^* x_{1l} \le h_{1i} && \forall i \in L \\
& \sum_j W_{2ij} y_{1j} - \sum_l TX_{2il} = h_{2i} && \forall i \in E \\
& \sum_j W_{1ij} y_{2j} - \sum_{l|Ind\_T_{1il}=1} T_{1il} X'_l - \sum_{l|Ind\_T_{1il}=0} T_{1il}^* X'_l + s_{1i} = h_{1i} && \forall i \in L \\
& \sum_j W_{2ij} y_{2j} - \sum_l T_{2il} X'_l = h_{2i} && \forall i \in E \\
& \sum_{i\in L} W_{1ij} w_{1i} + \sum_{i \in E} W_{2ij} w_{2i} - a_j = q_j && \forall j \text{ such that } Ind\_q_j = 1 \\
& \sum_{i\in L} W_{1ij} w_{1i} + \sum_{i \in E} W_{2ij} w_{2i} - a_j = q_j^* && \forall j \text{ such that } Ind\_q_j = 0 \\
& T_{2il}^L x_{1l} \le TX_{2il} \le T_{2il}^U x_{1l} && \forall i \in E, \forall l \\
& TX_{2il} \le T_{2il} - T_{2il}^L (1 - x_{1l}) && \forall i \in E, \forall l \\
& TX_{2il} \ge T_{2il} - T_{2il}^U (1 - x_{1l}) && \forall i \in E, \forall l \\
& w_{1i} s_{1i} = 0 && \forall i \in L \\
& a_j y_{2j} = 0 && \forall j \\
& T_{1il}^L \le T_{1il} \le T_{1il}^U && \forall i \in L\ \ \forall l \\
& T_{2il}^L \le T_{2il} \le T_{2il}^U && \forall i \in E\ \ \forall l \\
& h_{1i}^L \le h_{1i} \le h_{1i}^U && \forall i \in L \\
& h_{2i}^L \le h_{2i} \le h_{2i}^U && \forall i \in E \\
& y_{1j} \ge 0,\ y_{2j} \ge 0,\ s_{1i} \ge 0,\ a_j \ge 0,\ w_{1i} \ge 0,\ x_{1l} \in \{0,1\} && \forall i \in L, \forall j, \forall l
\end{aligned}
$$

Condition to add constraints	Constraint reference	Constraint index set
$Ind\_T_{1il} = 1$	(3.5.2.1.1)	For all $i \in L$, l
$Ind\_q_j = 1$	(3.5.2.1.2)	For all j

where

$$
Ind\_T_{1il} = \begin{cases} 1, & \text{if } T_{1il} \text{ value cannot be predetermined} \\ 0, & \text{otherwise} \end{cases}
$$

$$
Ind\_q_j = \begin{cases} 1, & \text{if } q_j \text{ value cannot be predetermined} \\ 0, & \text{otherwise} \end{cases}
$$

$$T_{1il}^{*}=\begin{cases}\text{Preprocessed value of } T_{1il}, \text{ if } T_{1il} \text{ can be preprocessed}\\ 0, \text{ Otherwise}\end{cases}$$

$$q_{j}^{*}=\begin{cases}\text{Preprocessed value of } q_{j}, \text{ if } q_{j} \text{ can be preprocessed}\\ 0, \text{ Otherwise}\end{cases}$$

For any branch and bound scheme, the branching rules are always critical. For the model (7), branching priorities are recommended as follows: (i) complementary slackness conditions, (ii) binary decisions on the parameters bounds, and (iii) first-stage binary decisions. Using this approach, the model (7) can be solved.

The optimal objective function value Δ^{U*} of the model (7) is used to update the value of Δ^{U} by setting Δ^{U} to min{Δ^{U*}, Δ^{U}}. The optimality condition is then checked. If the optimality condition is satisfied, the algorithm is terminated with an ε – optimal robust solution which is the discrete solution with the maximum regret of Δ^{U} from the model (7). Otherwise, add scenario $\omega_{X'}^{*}$ which is the combination of the optimal settings of $\bar{c}, T_1, T_2, \bar{h}_1, \bar{h}_2, \bar{q}, W_1, W_2$ from the model (7) to the scenario set Ω and set $S \leftarrow S - D$ such that $D = \{X \in S \mid \hat{\Delta}_X^U \geq \Delta^U\}$ and the second stage is repeated.

3.5.3 The Third Stage

The main purpose of the third stage algorithm is to identify a scenario $\omega_1^{*} \in \bar{\Omega}$ which admits no feasible solution to $Z_{\omega}^{*}(\bar{x}_{\Omega})$ for $\omega = \omega_1^{*}$. To achieve this goal, the algorithm solves a bi-level programming problem referred to as the bi-level-1 model by following two main steps. In the first step, the algorithm starts by pre-processing model parameters. At this point, some model parameters' values in the original bi-level-1 model are predetermined at their optimal setting by following some simple preprocessing rules. In the second step, the bi-level-1 model is transformed from its original form into a single-level mixed integer linear programming structure. Next we describe the key concepts of each algorithm step and the structure of the bi-level-1 model.

Recall that the scenario set $\bar{\Omega}$ is generated by all possible values of the parameter vector ξ. Let us define $\xi(\omega)$ as the specific setting of the parameter vector ξ under scenario $\omega \in \bar{\Omega}$ and $\Xi = \{\xi(\omega) \mid \omega \in \bar{\Omega}\}$ as the support of the random vector ξ. The following model (8) demonstrates the general structure of the bi-level-1 model

$$
\begin{aligned}
\min_{\xi} \quad & \delta \\
\text{s.t.} \quad & \xi \in \Xi \\
& \max_{\vec{y},\vec{s},\vec{s}_1,\vec{s}_2,\delta} \quad \delta \\
& \text{s.t.} \quad \boldsymbol{W}_1\vec{y} + \vec{s} = \vec{h}_1 + \boldsymbol{T}_1\vec{x}_{\Omega} \\
& \qquad \boldsymbol{W}_2\vec{y} + \vec{s}_1 = \vec{h}_2 + \boldsymbol{T}_2\vec{x}_{\Omega} \\
& \qquad -\boldsymbol{W}_2\vec{y} + \vec{s}_2 = -\vec{h}_2 - \boldsymbol{T}_2\vec{x}_{\Omega} \\
& \qquad \delta\vec{1} \le \vec{s}, \quad \delta\vec{1} \le \vec{s}_1, \quad \delta\vec{1} \le \vec{s}_2, \quad \vec{y} \ge \vec{0}
\end{aligned}
.
$$

In the bi-level-1 model, the leader's objective is to make the problem infeasible by controlling the parameters' settings. The follower's objective is to make the problem feasible by controlling the continuous decision variables, under the fixed parameters setting from the leader problem, when the setting of binary decision variables is fixed at $\vec{x}_{\Omega}$. In the model (8), δ represents a scalar decision variable and $\vec{0}$ and $\vec{1}$ represent the vector with all elements equal to zero and one respectively. The current form of the model (8) has a nonlinear bi-level structure with a set of constraints restricting the possible values of the decision vector $\xi = (\vec{c}, \vec{q}, \vec{h}_1, \vec{h}_2, \boldsymbol{T}_1, \boldsymbol{T}_2, \boldsymbol{W}_1, \boldsymbol{W}_2)$. Because the structure of the follower problem of the model (8) is a linear program and it affects the leader's decisions only through its objective function, we can simply replace this follower problem with explicit representations of its optimality conditions. These explicit representations include the follower's primal constraints, the follower's dual constraints and the follower's strong duality constraint.

Furthermore, from the special structure of the model (8), all elements in decision variable matrixes T_1, W_1 and vector $\vec{h}_1$ can be predetermined to either one of their bounds even before solving the model (8). For each element of the decision matrix W_1 in the model (8), the optimal setting of this decision variable is the upper bound of its possible values. The correctness of these simple rules is obvious based on the fact that $\vec{y} \geq \vec{0}$. Similarly, for each element of the decision vector $\vec{h}_1$ and matrix T_1, the optimal setting of this decision variable in the model (8) is the lower bound of its possible values.

***Lemma 3*:** (Assavapokee 2008b) The model (8) has at least one optimal solution $T_1^*, \vec{h}_1^*, W_1^*, T_2^*, \vec{h}_2^*$, and W_2^* in which each element of these vectors takes on a value at one of its bounds.

Proof: Because the optimal setting of each element of $T_1, \vec{h}_1$, and W_1 already takes its value from one of its bounds. We only need to prove this Lemma for each element of $T_2, \vec{h}_2$, and W_2. Each of these variables T_{2il}, h_{2i}, and W_{2ij} appears in only two constrains in the model (8) $\sum_j W_{2ij} y_j + s_{1i} = h_{2i} + \sum_l T_{2il} x_{\Omega l}$ and $-\sum_j W_{2ij} y_j + s_{2i} = -h_{2i} - \sum_l T_{2il} x_{\Omega l}$. It is also easy to see that $s_{1i} = -s_{2i}$ and $\min\{s_{1i}, s_{2i}\} = -|s_{1i} - s_{2i}|/2$. This fact implies that the optimal setting of $\vec{y}$ which maximizes $\min\{s_{1i}, s_{2i}\}$ will also minimize $|s_{1i} - s_{2i}|/2$ and vice versa under the fixed setting of ξ. Because $|s_{1i} - s_{2i}|/2 = |h_{2i} + \sum_l T_{2il}x_{\Omega l} - \sum_j W_{2ij}y_j|$, the optimal setting of T_{2il}, h_{2i}, and W_{2ij} will maximize $\min_{\vec{y}\in\chi(\vec{x}_\Omega)} |h_{2i} + \sum_l T_{2il}x_{\Omega l} - \sum_j W_{2ij}y_j|$ where $\chi(\vec{x}_\Omega) = \{\vec{y} \geq \vec{0} \mid \boldsymbol{W}_1\vec{y} \leq \vec{h}_1 + \boldsymbol{T}_1\vec{x}_\Omega,\ \boldsymbol{W}_2\vec{y} \leq \vec{h}_2 + \boldsymbol{T}_2\vec{x}_\Omega\}$. In this form, it is easy to see that the optimal setting of variables T_{2il}, h_{2i} and W_{2ij} will take on one of their bounds.

Let us define the notations L and E to represent sets of row indices associating with less-than-or-equal-to and equality constraints in the model (1) respectively. Let us also define the notations w_{1i} $\forall i \in L$, w_{2i}^+ and w_{2i}^- $\forall i \in E$ to represent dual variables of the follower problem in the model (8). Even though there are six sets of follower's constraints in the model (8), only three sets of dual variables are required. Because of the structure of dual constraints of the follower problem in the model (8), dual variables associated with the first three sets of the follower's constraints are exactly the same as those associated with the last three sets. After replacing the follower problem with explicit representations of its optimality conditions, we encounter with a number of nonlinear terms in the model including: $W_{2ij}y_j$, $W_{2ij}w_{2i}^+$, $W_{2ij}w_{2i}^-$, $h_{2i}w_{2i}^+$, $h_{2i}w_{2i}^-$, $T_{2il}w_{2i}^+$, and $T_{2il}w_{2i}^-$. By utilizing the result from Lemma 3, we can replace these nonlinear terms with the new set of variables WY_{2ij}, WW_{2ij}^+, WW_{2ij}^-, HW_{2i}^+, HW_{2i}^-, TW_{2il}^+, and TW_{2il}^- with the use of binary variables. We introduce binary variables biT_{2il}, bih_{2i}, and biW_{2ij} which take the value of zero or one if variables T_{2il}, h_{2i}, and W_{2ij} respectively take the lower or the upper bound values. The following three sets of equations 3.5.3.1, 3.5.3.2 and 3.5.3.3 will be used to relate these new variables with nonlinear terms in the model. In these constraints, the notations y_j^U , w_{2i}^{+U} and w_{2i}^{-U} represent the upper bound value of the variables y_j, w_{2i}^+ and w_{2i}^- respectively. Terlaky (1996) describes some techniques on constructing these bounds of the primal and dual variables.

$$\left.\begin{aligned}
&T_{2il} = T_{2il}^L + (T_{2il}^U - T_{2il}^L)biT_{2il} \\
&T_{2il}^L w_{2i}^+ \le TW_{2il}^+ \le T_{2il}^U w_{2i}^+ \\
&T_{2il}^L w_{2i}^- \le TW_{2il}^- \le T_{2il}^U w_{2i}^- \\
&TW_{2il}^+ \ge T_{2il}^U w_{2i}^+ - (|T_{2il}^U| + |T_{2il}^L|)(w_{2i}^{+U})(1 - biT_{2il}) \\
&TW_{2il}^+ \le T_{2il}^L w_{2i}^+ + (|T_{2il}^U| + |T_{2il}^L|)(w_{2i}^{+U})(biT_{2il}) \\
&TW_{2il}^- \ge T_{2il}^U w_{2i}^- - (|T_{2il}^U| + |T_{2il}^L|)(w_{2i}^{-U})(1 - biT_{2il}) \\
&TW_{2il}^- \le T_{2il}^L w_{2i}^- + (|T_{2il}^U| + |T_{2il}^L|)(w_{2i}^{-U})(biT_{2il}) \\
&\qquad biT_{2il} \in \{0,1\}
\end{aligned}\right\} \quad \forall i \in E,\ \forall l \qquad (3.5.3.1)$$

$$\left.\begin{aligned}
&h_{2i} = h_{2i}^L + (h_{2i}^U - h_{2i}^L)bih_{2i} \\
&h_{2i}^L w_{2i}^+ \le HW_{2i}^+ \le h_{2i}^U w_{2i}^+ \\
&h_{2i}^L w_{2i}^- \le HW_{2i}^- \le h_{2i}^U w_{2i}^- \\
&HW_{2i}^+ \ge h_{2i}^U w_{2i}^+ - (|h_{2i}^U| + |h_{2i}^L|)(w_{2i}^{+U})(1 - bih_{2i}) \\
&HW_{2i}^+ \le h_{2i}^L w_{2i}^+ + (|h_{2i}^U| + |h_{2i}^L|)(w_{2i}^{+U})(bih_{2i}) \\
&HW_{2i}^- \ge h_{2i}^U w_{2i}^- - (|h_{2i}^U| + |h_{2i}^L|)(w_{2i}^{-U})(1 - bih_{2i}) \\
&HW_{2i}^- \le h_{2i}^L w_{2i}^- + (|h_{2i}^U| + |h_{2i}^L|)(w_{2i}^{-U})(bih_{2i}) \\
&\qquad bih_{2i} \in \{0,1\}
\end{aligned}\right\} \quad \forall i \in E \qquad (3.5.3.2)$$

$$\left.\begin{array}{l}
W_{2ij} = W^L_{2ij} + (W^U_{2ij} - W^L_{2ij}) biW_{2ij} \\
W^L_{2ij} y_j \le WY_{2ij} \le W^U_{2ij} y_j \\
WY_{2ij} \ge W^U_{2ij} y_j - (|W^U_{2ij}| + |W^L_{2ij}|)(y^U_j)(1 - biW_{2ij}) \\
WY_{2ij} \le W^L_{2ij} y_j + (|W^U_{2ij}| + |W^L_{2ij}|)(y^U_j)(biW_{2ij}) \\
W^L_{2ij} w^+_{2i} \le WW^+_{2ij} \le W^U_{2ij} w^+_{2i} \\
W^L_{2ij} w^-_{2i} \le WW^-_{2ij} \le W^U_{2ij} w^-_{2i} \\
WW^+_{2ij} \ge W^U_{2ij} w^+_{2i} - (|W^U_{2ij}| + |W^L_{2ij}|)(w^{+U}_{2i})(1 - biW_{2ij}) \\
WW^+_{2ij} \le W^L_{2ij} w^+_{2i} + (|W^U_{2ij}| + |W^L_{2ij}|)(w^{+U}_{2i})(biW_{2ij}) \\
WW^-_{2ij} \ge W^U_{2ij} w^-_{2i} - (|W^U_{2ij}| + |W^L_{2ij}|)(w^{-U}_{2i})(1 - biW_{2ij}) \\
WW^-_{2ij} \le W^L_{2ij} w^-_{2i} + (|W^U_{2ij}| + |W^L_{2ij}|)(w^{-U}_{2i})(biW_{2ij}) \\
\qquad biW_{2ij} \in \{0,1\}
\end{array}\right\} \forall i \in E, \forall j \qquad (3.5.3.3)$$

After applying pre-processing rules, the follower problem transformation, and the result from Lemma 3, the model (8) can be transformed from a bi-level nonlinear structure to a single-level mixed integer linear structure presented in the model (9). The table in the model (9) is used to identify some additional constraints and conditions for adding these constraints to the model (9). These results greatly simplify the solution methodology of the Bi-level-1 model. If the optimal setting of the decision variable δ is negative, the algorithm will add scenario ω^*_1, which is generated by the optimal setting of $\vec{h}_1, \vec{h}_2, \boldsymbol{T}_1, \boldsymbol{T}_2, \boldsymbol{W}_1$, and $\boldsymbol{W}_2$ from the model (9) and any feasible combination of $\vec{c}$ and $\vec{q}$, to the scenario set Ω and return to the first stage algorithm. Otherwise the algorithm will forward the solution $\vec{x}_\Omega$ and the value of Δ^L to the fourth stage algorithm.

Model (9):

$$
\begin{aligned}
&\min \quad \delta \\
&\text{s.t.} \\
&\sum_j W_{1ij}^U y_j + s_i = h_{1i}^L + \sum_l T_{1il}^L x_{\Omega l} && \forall i \in L \\
&\sum_j WY_{2ij} + s_{1i} = h_{2i} + \sum_l T_{2il} x_{\Omega l} && \forall i \in E \\
&-\sum_j WY_{2ij} + s_{2i} = -h_{2i} - \sum_l T_{2il} x_{\Omega l} && \forall i \in E \\
&\delta \le s_i \quad \forall i \in L, \qquad \delta \le s_{1i} \quad \forall i \in E, \qquad \delta \le s_{2i} && \forall i \in E \\
&\sum_{i \in L} W_{1ij}^U w_{1i} + \sum_{i \in E} \left(WW_{2ij}^+ - WW_{2ij}^- \right) \ge 0 && \forall j \\
&\sum_{i \in L} w_{1i} + \sum_{i \in E} \left(w_{2i}^+ + w_{2i}^- \right) = 1 \\
&\delta = \sum_{i \in L} \left(h_{1i}^L + \sum_l T_{1il}^L x_{\Omega l} \right) w_{1i} + \sum_{i \in E} \left(HW_{2i}^+ - HW_{2i}^- + \sum_l \left(TW_{2il}^+ - TW_{2il}^- \right) x_{\Omega l} \right) \\
&w_{1i} \ge 0 \ \forall i \in L, \quad w_{2i}^+ \ge 0 \ \forall i \in E, \quad w_{2i}^- \ge 0 \ \forall i \in E, \quad y_j \ge 0 \ \forall j
\end{aligned}
$$

Condition to Add the Constraints	Constraint Reference	Constraint Index Set
Always	(3.5.3.1)	For all $i \in E$, For all l
Always	(3.5.3.2)	For all $i \in E$
Always	(3.5.3.3)	For all $i \in E$, For all j

3.5.4 The Fourth Stage

The main purpose of the fourth stage is to determine a new scenario ω' with the largest approximated regret value ($\hat{\Delta}_{X_\Omega}^U$) for the current candidate robust decision ($\vec{x}_\Omega$). Note that the approximation is performed in such a way that $\hat{\Delta}_{X_\Omega}^U \le \Delta_{X_\Omega}^U$.

In this thesis, we apply the Full Factorial approach developed by Assavapokee (2007a) as the approximation algorithm in finding the value of $\hat{\Delta}_{X_\Omega}^U$. The intuitive idea under the Full Factorial approach is as follow. Instead of searching for the scenario with maximum regret value through all possible points in the continuous interval for each uncertain parameter, we perform the search in a finite number of combinations of discrete points in each continuous interval. In other words, we discretize each compact interval into a finite set of discrete values that the parameters can take and solve the modified bi-level-2 model (bi-level-3 model) under this problem setting instead of solving the original bi-level-2 model. Figure 3.4 illustrates this discretization process.

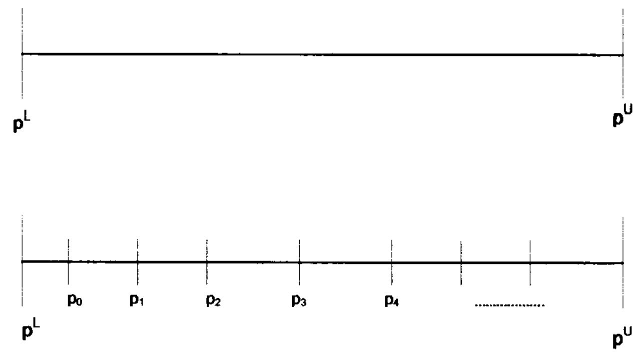


Figure 3.4 Discretization of the Compact Interval

The result from the bi-level-3 model is the approximated maximum regret value $\hat{\Delta}^{U}_{x_\Omega}$. Based on our numerical experiment, the computation time required to find the approximated maximum regret value using the bi-level-3 model is significantly less than the required computation time to find the maximum regret value using the bi-level-2 model. By using this type of approximation algorithms, the algorithm can quickly identify the quality of the given solution. We define the set $\Omega' \subset \bar{\Omega}$ as the finite set of all possible combinations of the discretized points.

In the bilevel-3 model, the leader's objective is to find the setting of the decision vector $\xi' = (\vec{c}, \vec{q}, \vec{h}_1, \vec{h}_2, T_1, T_2, W_1, W_2)$ and decision vector $(\vec{x}_1, \vec{y}_1)$ that result in the maximum regret value possible over all scenarios in the scenario set Ω', $\max_{\omega \in \Omega'} \{O^*_\omega - Z^*_\omega(\vec{x}_{\Omega'})\}$, for the candidate robust solution $\vec{x}_\Omega$. Let us define $\xi'(\omega)$ as the specific setting of the parameter vector ξ' under the scenario $\omega \in \Omega'$ and $\Xi' = \{\xi'(\omega) \mid \omega \in \Omega'\}$ as the support of the random vector ξ'. The follower's objective is to set the decision vector $\vec{y}_2$ to correctly calculate the value of $Z^{*}_{\omega}(\vec{x}_{\Omega})$ under the fixed setting of decision vector ξ' established by the leader. The general structure of the Bi-level-3 model is represented in the following model (10).

The solution methodology for solving the model (10) can be structured into two main steps. These two main steps include (1) the parameter pre-processing step and (2) the model transformation step. Each of these steps is described in detail in the following subsections

Model (10):

$$
\begin{aligned}
\max_{\vec{x}_1 \in \{0,1\}^{|\vec{x}_1|}, \vec{y}_1 \ge \vec{0}, \xi'} \quad & \{\vec{q}^T \vec{y}_1 + \vec{c}^T \vec{x}_1 - \vec{q}^T \vec{y}_2 - \vec{c}^T \vec{x}_{\Omega}\} \\
\text{s.t.} \quad & \xi' \in \Xi' \\
& \boldsymbol{W}_1 \vec{y}_1 \le \vec{h}_1 + \boldsymbol{T}_1 \vec{x}_1 \\
& \boldsymbol{W}_2 \vec{y}_1 = \vec{h}_2 + \boldsymbol{T}_2 \vec{x}_1 \\
& \max_{\vec{y}_2 \ge \vec{0}} \vec{q}^T \vec{y}_2 \\
& \text{s.t.} \quad \boldsymbol{W}_1 \vec{y}_2 \le \vec{h}_1 + \boldsymbol{T}_1 \vec{x}_{\Omega} \\
& \qquad\; \boldsymbol{W}_2 \vec{y}_2 = \vec{h}_2 + \boldsymbol{T}_2 \vec{x}_{\Omega}
\end{aligned}
$$

3.5.4.1 Parameter Pre-Processing Step

From the structure of the model (10), many elements of decision vector ξ' can be predetermined to attain their optimal setting at one of their bounds. In many cases, simple rules exist in identifying the optimal values of these elements of decision vector ξ' when the information on $\vec{x}_{\Omega}$ is given even before solving the model (10). The following section describes these simple pre-processing rules for elements of vector $\vec{c}$ and matrix T_1 in the vector ξ'.

Pre-Processing Step for $\vec{c}$

The elements of decision vector $\vec{c}$ represent the parameters corresponding to coefficients of binary decision variables in the objective function of the model (1).

Each element c_l of vector $\vec{c}$ is represented in the objective function of the model (10) as: $(c_l x_{1l} - c_l x_{\Omega l})$. From any given value of $x_{\Omega l}$, the value of c_l can be predetermined by the following simple rules. If $x_{\Omega l}$ is 1, the optimal setting of c_l is $c_l^* = c_l^L$. Otherwise the optimal setting of c_l is $c_l^* = c_l^U$.

Pre-Processing Step for T_1

The elements of decision vector T_1 represent the coefficients of the binary decision variables located in the less-than-or-equal-to constraints of the model (1). Each element T_{1il} of matrix T_1 is represented in the constraint of the model (10) as $\sum_j W_{1ij} y_{1j} \le h_{1i} + T_{1il} x_{1l} + \sum_{k \ne l} T_{1ik} x_{1k}$ and $\sum_j W_{1ij} y_{2j} \le h_{1i} + T_{1il} x_{\Omega l} + \sum_{k \ne l} T_{1ik} x_{\Omega k}$. From any given value of $x_{\Omega l}$, the value of T_{1il} can be predetermined at $T_{1il}^* = T_{1il}^U$ if $x_{\Omega l} = 0$. In the case when $x_{\Omega l} = 1$, the optimal setting of T_{1il} satisfies the following set of constraints illustrated in eq. 3.5.4.1.1 where the new variable TX_{11il} replaces the nonlinear term $T_{1il} x_{1l}$ in the model (10). The insight of this set of eq. 3.5.4.1.1 is that if the value of x_{1l} is set to be zero by the model, the optimal setting of T_{1il} is T_{1il}^L and $TX_{11il} = 0$. Otherwise the optimal setting of T_{1il} can not be predetermined and $TX_{11il} = T_{1il}$.

$$
\begin{aligned}
& TX_{11il} - T_{1il} + T_{1il}^L (1 - x_{1l}) \le 0 \\
& -TX_{11il} + T_{1il} - T_{1il}^U (1 - x_{1l}) \le 0 \\
& T_{1il}^L x_{1l} \le TX_{11il} \le T_{1il}^U x_{1l} \\
& T_{1il}^L \le T_{1il} \le T_{1il}^L + x_{1l} (T_{1il}^U - T_{1il}^L)
\end{aligned}
\qquad (3.5.4.1.1)
$$

3.5.4.2 Problem Transformation Step

In order to solve the model (10) efficiently, the following two main tasks have to be accomplished. First, a modeling technique is required to properly model the constraint $\xi' \in \Xi'$. Second, an efficient transformation method is required to transform the original formulation of the model (10) into a computationally efficient formulation. The following two subsections describe techniques and methodologies for performing these two tasks.

3.5.4.3 Modeling Technique for the Constraint $\xi' \in \Xi'$

Consider a variable p which only takes its value from $\bar{p}$ distinct real values, $p_{(1)}$, $p_{(2)}$, ..., $p_{(\bar{p})}$. This constraint on the variable p can be formulated in the mathematical programming model as: $p = \sum_{i=1}^{\bar{p}} p_{(i)} bi_i$, $\sum_{i=1}^{\bar{p}} bi_i = 1$, $bi_i \geq 0 \ \forall i = 1, ..., \bar{p}$ and $\{bi_1, bi_2, ..., bi_{\bar{p}}\}$ is SOS1. A Special Ordered Set of type One (SOS1) is defined to be a set of variables for which not more than one member from the set may be non-zero. When these nonnegative variables, $bi_i \ \forall i = 1, ..., \bar{p}$, are defined as SOS1, there are only $\bar{p}$ branches required in the searching tree for these variables.

3.5.4.4 Final Transformation Steps for the Bi-level-3 Model

Because the structure of the follower problem in the model (10) is a linear program and it affects the leader's decisions only through its objective function, the final transformation steps start by replacing the follower problem with explicit representations of its optimality conditions. These explicit representations include the follower's primal constraints, the follower's dual constraints and the follower's strong duality constraint. The model (11) illustrates the formulation of the model (10) after this first transformation where decision variables $w_{1i} \ \forall i \in L$ and $w_{2i} \ \forall i \in E$ represent the dual variable associated with follower's constraints. The model (11) is a single-level mixed integer nonlinear optimization problem. By applying results from parameter pre-processing steps and modeling technique previously discussed, the final transformation steps are completed and are summarized below

Model (11):

$$
\begin{aligned}
&\max\{\sum_j q_j y_{1j} + \sum_l c_l x_{1l} - \sum_j q_j y_{2j} - \sum_l c_l x_{\Omega l}\} \\
&\text{s.t.}\quad \xi' \in \Xi' \\
&\sum_j W_{1ij} y_{1j} \le h_{1i} + \sum_l T_{1il} x_{1l} \quad \forall i \in L \\
&\sum_j W_{1ij} y_{2j} \le h_{1i} + \sum_l T_{1il} x_{\Omega l} \quad \forall i \in L \\
&\sum_j W_{2ij} y_{1j} = h_{2i} + \sum_l T_{2il} x_{1l} \quad \forall i \in E \\
&\sum_j W_{2ij} y_{2j} = h_{2i} + \sum_l T_{2il} x_{\Omega l} \quad \forall i \in E \\
&\sum_{i \in L} W_{1ij} w_{1i} + \sum_{i \in E} W_{2ij} w_{2i} \ge q_j \quad \forall j \\
&\sum_{i \in L} \left(h_{1i} + \sum_l T_{1il} x_{\Omega l} \right) w_{1i} + \sum_{i \in E} \left(h_{2i} + \sum_l T_{2il} x_{\Omega l} \right) w_{2i} = \sum_j q_j y_{2j} \\
&w_{1i} \ge 0 \ \forall i \in L, \quad y_{1j} \ge 0, y_{2j} \ge 0 \ \forall j, \quad x_{1l} \in \{0,1\} \ \forall l
\end{aligned}
$$

Final Transformation Steps

Parameter $\bar{c}$: By applying the preprocessing rule, each variable c_l can be fixed at c_l^*.

Parameter T_1: By applying previous results, if the parameter T_{1il} can be preprocessed, then fix its value at the appropriate value of T_{1il}^*. Otherwise, first add a decision variable TX_{11il} and a set of constraints illustrated in eq. 3.5.4.1.1 to replace the nonlinear term $T_{1il}x_{1l}$ in the model (11), then add a set of variables and constraints illustrated in eq. 3.5.4.4.1 to replace part of the constraint $\xi' \in \Xi'$ for T_{1il} in the model (11), and finally add a variable TW_{1il} and a set of variables and constraints illustrated in eq. 3.5.4.4.2 to replace the nonlinear term $T_{1il}w_{1i}$ in the model (11), where w_{1i}^U and w_{1i}^L represent the upper bound and the lower bound of dual variable w_{1i} respectively

$$T_{1il} = \sum_{s=1}^{\bar{T}_{1il}} T_{1il(s)} biT_{1il(s)}, \sum_{s=1}^{\bar{T}_{1il}} biT_{1il(s)} = 1, \;\; biT_{1il(s)} \geq 0 \;\forall s \in \{1,2,...,\bar{T}_{1il}\} \text{ and } \bigcup_{\forall s} \{biT_{1il(s)}\} \text{ is SOS1} \quad (3.5.4.4.1)$$

$$\left. \begin{array}{l} TW_{1il} = \sum_{s=1}^{\bar{T}_{1il}} T_{1il(s)} ZTW_{1il(s)} \\ w_{1i}^{L} biT_{1il(s)} \leq ZTW_{1il(s)} \leq w_{1i}^{U} biT_{1il(s)}, \;\; ZTW_{1il(s)} \leq w_{1i} - w_{1i}^{L}(1 - biT_{1il(s)}), \\ ZTW_{1il(s)} \geq w_{1i} - w_{1i}^{U}(1 - biT_{1il(s)}) \;\; \forall s \in \{1,...,\bar{T}_{1il}\} \end{array} \right\} . \quad (3.5.4.4.2)$$

Parameter T_2: We first add a decision variable TX_{21il} and a set of constraints illustrated in eq. 3.5.4.4.3 to replace the nonlinear term $T_{2il}x_{1l}$ in the model (11), then add a set of variables and constraints illustrated in eq. 3.5.4.4.4 to replace part of the constraint $\xi' \in \Xi'$ for T_{2il} in the model (11), and finally add a variable TW_{2il} and a set of variables and constraints illustrated in eq. 3.5.4.4.5 to replace the nonlinear term $T_{2il}w_{2i}$ in the model (11), where w_{2i}^{U} and w_{2i}^{L} represent the upper bound and the lower bound of variable w_{2i} respectively

$$\left. \begin{array}{l} TX_{21il} = \sum_{s=1}^{\bar{T}_{2il}} T_{2il(s)} ZTX_{21il(s)} \\ 0 \leq ZTX_{21il(s)} \leq 1, \;\; ZTX_{21il(s)} \leq biT_{2il(s)}, \;\; ZTX_{21il(s)} \leq x_{1i}, \\ ZTX_{21il(s)} \geq biT_{2il(s)} + x_{1i} - 1 \;\; \forall s \in \{1,...,\bar{T}_{2il}\} \end{array} \right\} \quad (3.5.4.4.3)$$

$$T_{2il} = \sum_{s=1}^{\bar{T}_{2il}} T_{2il(s)} biT_{2il(s)}, \;\; \sum_{s=1}^{\bar{T}_{2il}} biT_{2il(s)} = 1, \;\; biT_{2il(s)} \geq 0 \;\forall s \in \{1,2,...,\bar{T}_{2il}\} \text{ and } \bigcup_{\forall s} \{biT_{2il(s)}\} \text{ is SOS1} \quad (3.5.4.4.4)$$

$$\left.\begin{array}{l} TW_{2il} = \sum_{s=1}^{\bar{T}_{2il}} T_{2il(s)} ZTW_{2il(s)} \\ w_{2i}^{L} biT_{2il(s)} \le ZTW_{2il(s)} \le w_{2i}^{U} biT_{2il(s)},\ ZTW_{2il(s)} \le w_{2i} - w_{2i}^{L}(1 - biT_{2il(s)}), \\ ZTW_{2il(s)} \ge w_{2i} - w_{2i}^{U}(1 - biT_{2il(s)})\ \ \forall s \in \{1,\ldots,\bar{T}_{2il}\} \end{array}\right\}. \tag{3.5.4.4.5}$$

Parameter $\vec{h}_1$ and $\vec{h}_2$: We first add a set of variables and constraints illustrated in equations 3.5.4.4.6 and 3.5.4.4.7 to replace part of the constraint $\xi' \in \Xi'$ for h_{1i} and h_{2i} respectively in the model (11). We then add variables HW_{1i}, HW_{2i} and a set of variables and constraints in equations 3.5.4.4.8 and 3.5.4.4.9 to replace the nonlinear terms $h_{1i}w_{1i}$ and $h_{2i}w_{2i}$ respectively in the model (11)

$$h_{1i} = \sum_{s=1}^{\bar{h}_{1i}} h_{1i(s)} biH_{1i(s)},\ \sum_{s=1}^{\bar{h}_{1i}} biH_{1i(s)} = 1,\ biH_{1i(s)} \ge 0\ \forall s \in \{1,2,\ldots,\bar{h}_{1i}\} \tag{3.5.4.4.6}$$

and $\bigcup_{\forall s}\{biH_{1i(s)}\}$ is SOS1

$$h_{2i} = \sum_{s=1}^{\bar{h}_{2i}} h_{2i(s)} biH_{2i(s)},\ \sum_{s=1}^{\bar{h}_{2i}} biH_{2i(s)} = 1,\ biH_{2i(s)} \ge 0\ \forall s \in \{1,2,\ldots,\bar{h}_{2i}\} \tag{3.5.4.4.7}$$

and $\bigcup_{\forall s}\{biH_{2i(s)}\}$ is SOS1

$$\left.\begin{array}{l} HW_{1i} = \sum_{s=1}^{\bar{h}_{1i}} h_{1i(s)} ZHW_{1i(s)} \\ w_{1i}^{L} biH_{1i(s)} \le ZHW_{1i(s)} \le w_{1i}^{U} biH_{1i(s)},\ ZHW_{1i(s)} \le w_{1i} - w_{1i}^{L}(1 - biH_{1i(s)}), \\ ZHW_{1i(s)} \ge w_{1i} - w_{1i}^{U}(1 - biH_{1i(s)})\ \ \forall s \in \{1,\ldots,\bar{h}_{1i}\} \end{array}\right\} \tag{3.5.4.4.8}$$

$$\left.\begin{array}{l} HW_{2i} = \sum_{s=1}^{\bar{h}_{2i}} h_{2i(s)} ZHW_{2i(s)} \\ w_{2i}^{L} biH_{2i(s)} \le ZHW_{2i(s)} \le w_{2i}^{U} biH_{2i(s)},\ ZHW_{2i(s)} \le w_{2i} - w_{2i}^{L}(1 - biH_{2i(s)}), \\ ZHW_{2i(s)} \ge w_{2i} - w_{2i}^{U}(1 - biH_{2i(s)})\ \ \forall s \in \{1,\ldots,\bar{h}_{2i}\} \end{array}\right\}. \tag{3.5.4.4.9}$$

Parameter $\vec{q}$: We first add a set of variables and constraints illustrated in eq. 3.5.4.4.10 to replace part of the constraint $\xi' \in \Xi'$ for q_j in the model (11). We then add decision variables QY_{1j}, QY_{2j} and a set of variables and constraints in eq.

3.5.4.4.11 to replace the nonlinear terms $q_j y_{1j}$ and $q_j y_{2j}$ respectively in the model (11) where y_{rj}^U and y_{rj}^L represent the upper bound and the lower bound of variable y_{rj} respectively for $r = 1$ and 2

$$q_j = \sum_{s=1}^{\bar{q}_j} q_{j(s)} biQ_{j(s)},\ \sum_{s=1}^{\bar{q}_j} biQ_{j(s)} = 1,\ biQ_{j(s)} \geq 0\ \forall s \in \{1,2,...,\bar{q}_j\} \quad (3.5.4.4.10)$$

and $\bigcup_{\forall s}\{biQ_{j(s)}\}$ is SOS1

$$\left.\begin{array}{l} QY_{1j} = \sum_{s=1}^{\bar{q}_j} q_{j(s)} ZQY_{1j(s)} \\ y_{1j}^L biQ_{j(s)} \leq ZQY_{1j(s)} \leq y_{1j}^U biQ_{j(s)},\ ZQY_{1j(s)} \leq y_{1j} - y_{1j}^L(1 - biQ_{j(s)}), \\ ZQY_{1j(s)} \geq y_{1j} - y_{1j}^U(1 - biQ_{j(s)})\ \ \forall s \in \{1,...,\bar{q}_j\} \\ QY_{2j} = \sum_{s=1}^{\bar{q}_j} q_{j(s)} ZQY_{2j(s)} \\ y_{2j}^L biQ_{j(s)} \leq ZQY_{2j(s)} \leq y_{2j}^U biQ_{j(s)},\ ZQY_{2j(s)} \leq y_{2j} - y_{2j}^L(1 - biQ_{j(s)}), \\ ZQY_{2j(s)} \geq y_{2j} - y_{2j}^U(1 - biQ_{j(s)})\ \ \forall s \in \{1,...,\bar{q}_j\} \end{array}\right\} . \quad (3.5.4.4.11)$$

Parameter W_1 and W_2: We first add a set of variables and constraints illustrated in equations 3.5.4.4.12 and 3.5.4.4.13 to replace part of the constraint $\xi' \in \Xi'$ for W_{1ij} and W_{2ij} respectively in the model (11). We then add a set of variables and constraints illustrated in equations 3.5.4.4.14 and 3.5.4.4.15 together with variables $WY_{11ij}, WY_{21ij}, WY_{12ij}, WY_{22ij}, WW_{1ij}$, and WW_{2ij} to replace the nonlinear terms $W_{1ij} y_{1j}$, $W_{2ij} y_{1j}$ $W_{1ij} y_{2j}$, $W_{2ij} y_{2j}, W_{1ij} w_{1i}$, and $W_{2ij} w_{2i}$ in the model (11)

$$W_{1ij} = \sum_{s=1}^{\bar{W}_{1ij}} W_{1ij(s)} biW_{1ij(s)}, \ \sum_{s=1}^{\bar{W}_{1ij}} biW_{1ij(s)} = 1, \ biW_{1ij(s)} \geq 0 \ \forall s \in \{1,2,...,\bar{W}_{1ij}\} \qquad (3.5.4.4.12)$$

and $\bigcup_{\forall s}\{biW_{1ij(s)}\}$ is SOS1

$$W_{2ij} = \sum_{s=1}^{\bar{W}_{2ij}} W_{2ij(s)} biW_{2ij(s)}, \ \sum_{s=1}^{\bar{W}_{2ij}} biW_{2ij(s)} = 1, \ biW_{2ij(s)} \geq 0 \ \forall s \in \{1,2,...,\bar{W}_{2ij}\} \qquad (3.5.4.4.13)$$

and $\bigcup_{\forall s}\{biW_{2ij(s)}\}$ is SOS1

$$\left.\begin{array}{l}
WY_{11ij} = \sum_{s=1}^{\bar{W}_{1ij}} W_{1ij(s)} ZWY_{11ij(s)} \\
y_{1j}^{L} biW_{1ij(s)} \leq ZWY_{11ij(s)} \leq y_{1j}^{U} biW_{1ij(s)}, \ ZWY_{11ij(s)} \leq y_{1j} - y_{1j}^{L}(1 - biW_{1ij(s)}), \\
ZWY_{11ij(s)} \geq y_{1j} - y_{1j}^{U}(1 - biW_{1ij(s)}) \quad \forall s \in \{1,...,\bar{W}_{1ij}\} \\
WY_{12ij} = \sum_{s=1}^{\bar{W}_{1ij}} W_{1ij(s)} ZWY_{12ij(s)} \\
y_{2j}^{L} biW_{1ij(s)} \leq ZWY_{12ij(s)} \leq y_{2j}^{U} biW_{1ij(s)}, \ ZWY_{12ij(s)} \leq y_{2j} - y_{2j}^{L}(1 - biW_{1ij(s)}), \\
ZWY_{12ij(s)} \geq y_{2j} - y_{2j}^{U}(1 - biW_{1ij(s)}) \quad \forall s \in \{1,...,\bar{W}_{1ij}\} \\
WW_{1ij} = \sum_{s=1}^{\bar{W}_{1ij}} W_{1ij(s)} ZWW_{1ij(s)} \\
w_{1i}^{L} biW_{1ij(s)} \leq ZWW_{1ij(s)} \leq w_{1i}^{U} biW_{1ij(s)}, \ ZWW_{1ij(s)} \leq w_{1i} - w_{1i}^{L}(1 - biW_{1ij(s)}), \\
ZWW_{1ij(s)} \geq w_{1i} - w_{1i}^{U}(1 - biW_{1ij(s)}) \quad \forall s \in \{1,...,\bar{W}_{1ij}\}
\end{array}\right\} \qquad (3.5.4.4.14)$$

$$\left.\begin{array}{l}
WY_{21ij} = \sum_{s=1}^{\bar{W}_{2ij}} W_{2ij(s)} ZWY_{21ij(s)} \\
y_{1j}^{L} biW_{2ij(s)} \leq ZWY_{21ij(s)} \leq y_{1j}^{U} biW_{2ij(s)}, \ ZWY_{21ij(s)} \leq y_{1j} - y_{1j}^{L}(1 - biW_{2ij(s)}), \\
ZWY_{21ij(s)} \geq y_{1j} - y_{1j}^{U}(1 - biW_{2ij(s)}) \quad \forall s \in \{1,...,\bar{W}_{2ij}\} \\
WY_{22ij} = \sum_{s=1}^{\bar{W}_{2ij}} W_{2ij(s)} ZWY_{22ij(s)} \\
y_{2j}^{L} biW_{2ij(s)} \leq ZWY_{22ij(s)} \leq y_{2j}^{U} biW_{2ij(s)}, \ ZWY_{22ij(s)} \leq y_{2j} - y_{2j}^{L}(1 - biW_{2ij(s)}), \\
ZWY_{22ij(s)} \geq y_{2j} - y_{2j}^{U}(1 - biW_{2ij(s)}) \quad \forall s \in \{1,...,\bar{W}_{2ij}\} \\
WW_{2ij} = \sum_{s=1}^{\bar{W}_{2ij}} W_{2ij(s)} ZWW_{2ij(s)} \\
w_{2i}^{L} biW_{2ij(s)} \leq ZWW_{2ij(s)} \leq w_{2i}^{U} biW_{2ij(s)}, \ ZWW_{2ij(s)} \leq w_{2i} - w_{2i}^{L}(1 - biW_{2ij(s)}), \\
ZWW_{2ij(s)} \geq w_{2i} - w_{2i}^{U}(1 - biW_{2ij(s)}) \quad \forall s \in \{1,...,\bar{W}_{2ij}\}
\end{array}\right\}. \qquad (3.5.4.4.15)$$

By applying these transformation steps, the model (11) can now be transformed into its final formulation as a single level mixed integer linear programming problem as shown in the model (12). The table in the model eq. 3.5.4.4.16 is used to identify some additional constraints and conditions for adding these constraints to the model (12):

$$\Delta^{U^*} = \max\{\sum_j QY_{1j} + \sum_l c_l^* x_{1l} - \sum_j QY_{2j} - \sum_l c_l^* x_{\Omega l}\}$$

$$\text{s.t.} \quad \sum_j WY_{11ij} \le h_{1i} + \sum_{l \mid Ind_{il}=1} TX_{11il} + \sum_{l \mid Ind_{il}=0} T_{1il}^* x_{1l} \quad \forall i \in L$$

$$\sum_j WY_{21ij} = h_{2i} + \sum_l TX_{21il} \quad \forall i \in E$$

$$\sum_j WY_{12ij} \le h_{1i} + \sum_{l \mid Ind_{il}=1} T_{1il} x_{\Omega l} + \sum_{l \mid Ind_{il}=0} T_{1il}^* x_{\Omega l} \quad \forall i \in L$$

$$\sum_j WY_{22ij} = h_{2i} + \sum_l T_{2il} x_{\Omega l} \quad \forall i \in E$$

$$\sum_{i \in L} WW_{1ij} + \sum_{i \in E} WW_{2ij} \ge q_j \quad \forall j$$

$$\sum_{i \in L} HW_{1i} + \sum_{i \in E} HW_{2i} + \sum_{i \in L}\left(\sum_{l \mid Ind_{il}=1} TW_{1il} x_{\Omega l} + \sum_{l \mid Ind_{il}=0} T_{1il}^* w_{1i} x_{\Omega l} \right) + \sum_{i \in E}\left(\sum_l TW_{2il} x_{\Omega l} \right) = \sum_j QY_{2j}$$

$$y_{1j} \ge 0, y_{2j} \ge 0 \ \forall j, \quad w_{1i} \ge 0 \ \forall i \in L, \ x_{1l} \in \{0,1\} \ \forall l$$

Condition to Add the Constraints	Constraint Reference	Constraint Index Set
$Ind_{il} = 1$	3.5.4.1.1, 3.5.4.4.1, 3.5.4.4.2	For all $i \in L$, For all l
Always	3.5.4.4.3, 3.5.4.4.4, 3.5.4.4.5	For all $i \in E$, For all l
Always	3.5.4.4.6, 3.5.4.4.8	For all $i \in L$
Always	3.5.4.4.7, 3.5.4.4.9	For all $i \in E$
Always	3.5.4.4.10, 3.5.4.4.11	For all j
Always	3.5.4.4.12, 3.5.4.4.14	For all $i \in L$, For all j
Always	3.5.4.4.13, 3.5.4.4.15	For all $i \in E$, For all j

Where

$$Ind_{il} = \begin{cases} 1 & \text{if } T_{1il} \text{ value cannot be predetermined.} \\ 0 & \text{otherwise} \end{cases}$$

$$T_{1il}^{*} = \begin{cases} \text{Preprocessed value of } T_{1il} \text{ if } T_{1il} \text{ can be preprocessed} \\ 0 \quad \text{Otherwise} \end{cases}$$

The bi-level-3 model can now be solved as a mixed integer linear program by solving the model (12). The optimal objective function value of the model (12), $\hat{\Delta}^U_{X_\Omega}$, is used to determine the quality of the solution $\bar{x}_\Omega$. In addition, we do not have to solve the model (12) to optimality in every iteration to generate the scenario ω'. We can stop the optimization process for the model (12) as soon as the feasible solution with the objective value larger than the current value of Δ^U has been found. We can use the resulting feasible setting of $\xi' = (\bar{c}, \bar{q}, \bar{h}_1, \bar{h}_2, T_1, T_2, W_1, W_2)'$ to generate the scenario ω' in the current iteration.

After the bi-level-3 model is solved, if $\hat{\Delta}^U_{X_\Omega} \geq \Delta^U$, the generated scenario is added to the scenario set Ω and the first stage is repeated. Otherwise, the candidate robust solution $\bar{x}_\Omega$ is added to S. The algorithm then checks whether the candidate robust solution is a promising potential solution. The solution $\bar{x}_\Omega$ is considered promising if $\Delta^U > \hat{\Delta}^U_{\bar{x}_\Omega} > \varepsilon + \Delta^L$.

If the solution $\bar{x}_\Omega$ is a promising potential solution, it will be removed from the set S and the bi-level-2 model is solved for this solution. The new generated scenario is then added to the scenario set Ω and the Δ^U value is updated, $\Delta^U \leftarrow \min\{\Delta^U_{X_\Omega}, \Delta^U\}$. By using the Δ^U and Δ^L information, the optimality condition can be checked ($\Delta^U - \Delta^L \leq \varepsilon$). If the optimality condition is satisfied, the algorithm terminates with the solution associated with the maximum regret value of Δ^U as a ε-optimal robust solution. Otherwise, the algorithm updates S by deleting any solution, X, such that $\hat{\Delta}_X^U \geq \Delta^U$ and the first stage is repeated.

The following Lemma 4 provides the important result that the proposed four-stage algorithm always terminates at an ε-optimal robust solution in a finite number of algorithm steps.

***Lemma 4*:** The four-stage algorithm terminates in a finite number of steps. When the algorithm has terminated with $\varepsilon \geq 0$, it has either detected infeasibility or has found a ε-optimal robust solution to the original problem.

Proof: The result follows from the proof in Assavapokee (2007a, 2008b) and the definition of Δ^U and Δ^L and the fact that in the worst case the algorithm enumerates all scenarios $\omega \in \bar{\Omega}$ and $|\bar{\Omega}|$ is finite.

Figure 3.5 details the information flow of the proposed algorithm.

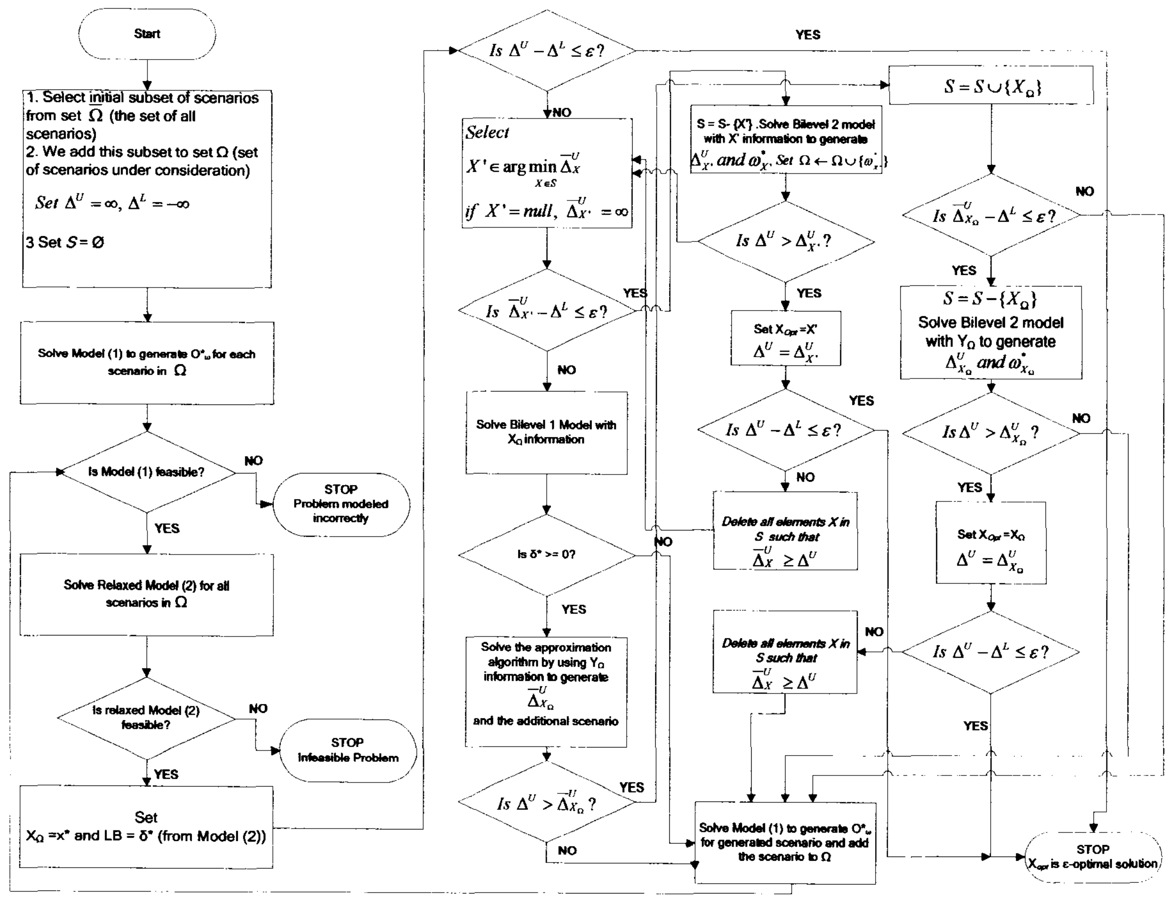


Figure 3.5: Flow Chart of Four-Stage Algorithm

In the next chapter, we apply the 4 stage algorithm to solve a facility location supply chain problem under interval data uncertainty.

Chapter 4

CASE STUDY

4.1 Introduction

In this chapter, we apply the proposed four-stage algorithm to a hypothetical robust supply chain facility location problem with infinite number of possible scenarios that rules out an extensive scheme. We consider a supply chain in which suppliers send material to factories that supply warehouses that supply markets as shown in Figure 4.1 (Chopra and Meindl, 2003). Location and capacity allocation decisions have to be made for both factories and warehouses. Multiple warehouses may be used to satisfy demand at a market and multiple factories may be used to replenish warehouses. It is also assumed that units have been appropriately adjusted such that one unit of input from a supply source produces one unit of the finished product. In addition, each factory and each warehouse cannot operate at more than its capacity and a linear penalty cost is incurred for each unit of unsatisfied demand. The model requires the following inputs:

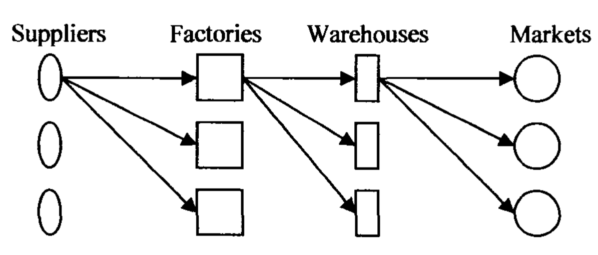


Figure 4.1: Stages in the Supply Chain Network (Chopra and Meindl, 2003).

In the deterministic case, the goal is to identify factory and warehouse locations as well as quantities shipped between various points in the supply chain that minimize the total fixed and variable costs. In this case, the overall problem can be modeled as the mixed integer linear programming problem presented in the following model (13):

Table 4.1: Description of Model Parameters and Variables

m	: Number of markets	W_e	: Potential warehouse capacity at site e
n	: Number of potential factory location	f_{1i}	: Fixed cost of locating a plant at site i
l	: Number of suppliers	f_{2e}	: Fixed cost of locating a warehouse at site e
t	: Number of potential warehouse locations	c_{1hi}	: Cost of shipping one unit from supplier h to factory i
D_j	: Annual demand from customer j	c_{2ie}	: Cost of shipping one unit from factory to warehouse e
K_i	: Potential capacity of factory site i	c_{3ej}	: Cost of shipping one unit from warehouse e to market j
S_h	: Supply capacity at supplier h	p_j	: Penalty cost per unit of unsatisfied demand at market j
x_i	: = 1 if plant is opened at site i; : = 0 otherwise	z_e	: = 1 if warehouse is opened at site e; : = 0 otherwise
x_{1hi}	: = Transportation quantity from supplier h to plant i	x_{2ie}	: = Transportation quantity from plant i to warehouse e
x_{3ej}	: = Transportation quantity from warehouse e to market j	s_j	: = Quantity of unsatisfied Demand at market j

$$\min \quad \sum_{i=1}^{n} f_{1i}x_i + \sum_{e=1}^{t} f_{2e}z_e + \sum_{h=1}^{l}\sum_{i=1}^{n} c_{1hi}y_{1hi} + \sum_{i=1}^{n}\sum_{e=1}^{t} c_{2ie}y_{2ie} + \sum_{e=1}^{t}\sum_{j=1}^{m} c_{3ej}y_{3ej} + \sum_{j=1}^{m} p_j s_j$$

$$s.t. \quad \sum_{i=1}^{n} y_{1hi} \le S_h \quad \forall h \in \{1,\ldots,l\}, \quad \sum_{h=1}^{l} y_{1hi} - \sum_{e=1}^{t} y_{2ie} = 0 \quad \forall i \in \{1,\ldots,n\}$$

$$\sum_{e=1}^{t} y_{2ie} \le K_i x_i \quad \forall i \in \{1,\ldots,n\}, \quad \sum_{i=1}^{n} y_{2ie} - \sum_{j=1}^{m} y_{3ej} = 0 \quad \forall e \in \{1,\ldots,t\}$$

$$\sum_{j=1}^{m} y_{3ej} \le W_e z_e \quad \forall e \in \{1,\ldots,t\}, \quad \sum_{e=1}^{t} y_{3ej} + s_j = D_j \quad \forall j \in \{1,\ldots,m\}$$

$$y_{1hi} \ge 0,\ y_{2ie} \ge 0,\ y_{3ej} \ge 0,\ s_j \ge 0,\ x_i \in \{0,1\},\ \text{and}\ z_e \in \{0,1\}$$

4.2 Case Study

When some parameters in this model are uncertain (ambiguous), the goal becomes to identify robust factory and warehouse locations (long term decisions) with an objective utilizing the deviation robust definition. Transportation decisions (short term decisions) are now recourse decisions which can be made after all model parameters' values are realized. We consider 9 possible setting of the parameters as summarized in the table below.

Table 4.2: Test Problem settings

Problem set	*l(suppliers)*	*n(factories)*	*t(warehouses)*	*m(markets)*
1	6	6	6	6
2	6	6	6	6
3	6	6	6	6
4	6	6	6	6
5	8	8	8	8
6	8	8	8	8
7	8	8	8	8
8	8	8	8	8
9	8	8	8	8

Each problem in the set above contains different sets of uncertain parameters and different sets of possible locations which can result in a large number of possible scenarios. The key uncertain parameters in these problems are the supply quantity at the supplier, the potential capacity at the factory, the potential capacity at the warehouse, and the unit penalty cost for not meeting demand at the market.

We assume that each uncertain (ambiguous) parameter can take its values from the real compact interval between 80% and 120% of its approximated value. The distance between each pair of locations is calculated based on the latitude and the longitude of each location.

The case study is solved on a Windows XP-based Pentium Dual Core CPU 2.00 GHz personal computer with 2.00 GB RAM using a GAMS optimization program together with CPLEX 10 for the optimization process.

4.3 Case study results

We apply the proposed algorithm to the 9 test problems shown in Table 4.2 by considering two different settings of initial scenarios. In the first setting, we consider only one scenario in the starting scenario set. In the second setting, we consider 16 scenarios in the starting scenario set. The 16 starting scenarios are obtained by considering the combinations of upper and lower bounds for each type of uncertain parameters. We have four main types of uncertain parameters in the problem set. Hence the starting scenario set of these problems consists of 16 (2^4) scenarios. The different case studies we run to gauge the performance of the four-stage algorithm are detailed in Table 4.3.

Table 4.3: Case study settings

Case study	$\lvert\Omega\rvert$	Benders decomposition	4 stage algorithm
1	1	Yes	Yes
2	16	Yes	Yes
3	1	No	Yes
4	16	No	Yes
5	1	No	No
6	16	No	No

We can now summarize the various insights observed from the computational effort. All results illustrate the significant improvement in computation time of the proposed algorithm over the existing methods. They also illustrate the applicability and effectiveness of our four-stage algorithm for solving large scale min-max regret robust optimization problems that have a mixed integer (binary) linear programming base model with interval data uncertainty.

The following Tables 4.4 & 4.5 summarizes the performance by using the new 4 stage algorithm. for both the 1 starting scenario and the 16 starting scenarios cases.

Table 4.4: Performance of 4 stage algorithm with 1 starting scenario

	|Ω|=1				
	Time in seconds				
Problem set	Bender	App time	Bi-level 2	Overall	# of scns/itns
1	29.47	33.7	146.213	209.383	29/28
2	14.86	1469.79	264	1748.65	18/17
3	12.13	24.5	216.51	253.14	16/15
4	25.29	1702.35	1077.45	2805.09	23/22
5	137.44	142.08	3704	3983.52	15/14
6	550.35	26552.44	36623.13	63725.92	43/42
7	221.2	948	29173.14	30342.34	20/19
8	548.45	77.13	8655.52	9281.1	36/35
9	952.1	12816.18	38396.64	52164.92	49/48

Table 4.5: Performance of 4 stage algorithm with 16 starting scenarios

	|Ω|=16				
	Time in seconds				
Problem set	Bender	App time	Bi-level 2	Overall	# of scns/itns
1	10.04	15.7	146.213	171.953	26/10
2	6.19	1310.51	264	1580.7	27/11
3	6.85	9.01	216.51	232.37	21/5
4	5.31	1201.11	1077.45	2283.87	24/8
5	37.61	178.697	3704	3920.307	22/6
6	149.85	24394.77	36623.13	61167.75	33/17
7	39.83	1069.56	29173.14	30282.53	26/10
8	158.36	36.21	8655.52	8850.09	27/11
9	110.94	8125.81	38396.64	46633.39	31/15

The following Tables 4.6 & 4.7 summarizes the performance by using the 4 stage algorithm without the use of Benders decomposition in the first stage. The notation MILP2 in the following tables represents solving the Model (2) without Benders' decomposition.

Table 4.6: Performance without Benders decomposition and with Approximation algorithm and 1 starting scenario

	|Ω|=1				
	Time in seconds				
Problem set	MILP2	App time	Bi-level 2	Overall	# of scns/itns
1	237	33.7	146.21	416.91	28/27
2	39.92	1469.79	264	1773.71	17/16
3	42.85	24.5	216.51	283.86	16/15
4	149.1	1702.35	1077.45	2928.9	23/22
5	388.43	142.08	3704	4234.51	15/14
6	7782.77	26552.44	36623.13	70958.34	43/42
7	840.37	948	29173.14	30961.51	20/19
8	4630.19	77.13	8655.52	13362.84	36/35
9	17509.4	12816.18	38396.64	68722.22	48/47

Table 4.7: Performance without Benders decomposition and with Approximation algorithm and 16 starting scenarios

	|Ω|=16				
	Time in seconds				
Problem set	MILP2	App time	Bi-level 2	Overall	# of scns/itns
1	75.08	15.7	146.21	236.99	26/10
2	40.61	1310.51	264	1615.12	27/11
3	22.59	9.01	216.51	248.11	21/5
4	22.79	1201.11	1077.45	2301.35	24/8
5	119.7	178.69	3704	4002.41	22/6
6	918.3	24394.77	36623.13	61936.17	33/17
7	322.4	1069.56	29173.14	30565.08	26/10
8	748.5	36.21	8655.52	9440.21	27/11
9	1089	8125.81	38396.64	47611.37	31/15

The following Tables 4.8 & 4.9 summarizes the performance by using the old 3 stage algorithm.

Table 4.8: Performance using the 3-stage algorithm with 1 starting scenario

	|Ω|=1			
	Time in seconds			
Problem set	MILP2	Bi-level 2	Overall	# of scns/itns
1	151.59	784.06	935.65	21/20
2	52.41	7929.68	7982.09	18/17
3	37.69	1396.37	1434.06	15/14
4	165.82	6532.56	6698.38	23/22
5	1009.11	28048.43	29057.54	25/24
6	N/A	N/A	N/A	N/A
7	3418.18	112316.3	115734.5	32/31
8	7141.49	93430.95	100572.4	41/40
9	28747.36	328046.77	356794.1	58/57

Table 4.9: Performance using the old 3 stage algorithm with 16 starting scenarios

	|Ω|=16			
	Time in seconds			
Problem set	MILP2	Bi-level 2	Overall	# of scns/itns
1	76.15	798.06	874.21	26/10
2	45.25	6856.32	6901.57	27/11
3	22.46	1097.44	1119.9	21/5
4	22.62	6220.5	6243.12	24/8
5	117.43	26615.26	26732.69	22/6
6	1104.25	1086922.56	1088026.81	34/18
7	319.45	105732.52	106051.97	26/10
8	761.25	89832.8	90594.05	27/11
9	1103.44	172808.52	173911.96	31/15

The following tables 4.10 & 4.11 summarize the improvement in computation times between the 3-stage algorithm and the new 4-stage algorithm.

Table 4.10: Improvement of 4 stage algorithm over 3-stage algorithm with 1 starting scenario

	|Ω|=1			
	New Algorithm		Old Algorithm	Reduction in computation time (%)
Problem set	Without Bender	With Bender		
1	416.913	209.383	935.65	77.62%
2	1773.71	1748.65	7982.09	78.09%
3	283.86	253.14	1434.06	82.35%
4	2928.9	2805.09	6698.38	58.12%
5	4234.51	3983.52	29057.54	86.29%
6	70958.34	63725.92	N/A	N/A
7	30961.51	30342.34	115734.5	73.78%
8	13362.84	9281.1	100572.4	90.77%
9	68722.22	52164.92	356794.1	85.38%

Table 4.11: Improvement of 4 stage algorithm over 3-stage algorithm with 16 starting scenarios

	|Ω|=16			
	New Algorithm		Old Algorithm	Reduction in computation time (%)
Problem set	Without Bender	With Bender		
1	236.993	171.953	874.21	80.33%
2	1615.12	1580.7	6901.57	77.10%
3	248.11	232.37	1119.9	79.25%
4	2301.35	2283.87	6243.12	63.42%
5	4002.417	3920.307	26732.69	85.34%
6	61936.17	61167.75	1088026.81	94.38%
7	30565.08	30282.53	106051.97	71.45%
8	9440.21	8850.09	90594.05	90.23%
9	47611.37	46633.39	173911.96	73.19%

In conclusion the tables 4.10 & 4.11 illustrate that there are significant improvements in computational performance between the old and the new algorithms. The minimum improvement from all cases is 58.12%. The maximum improvement from all cases is 94.38%. The average value of the percentage improvement from all cases is 79.24%. Further ways to improve the algorithms performance are mentioned in Chapter 5.

Chapter 5

SUMMARY AND FUTURE RESEARCH DIRECTION

5.1 Summary of Research

In this research, we develop a new min-max regret robust optimization algorithm for two-stage mixed integer (binary) linear programming problems under interval data uncertainty based on the original work by Assavapokee (2004, 2007a, 2008b). The algorithm is designed explicitly to handle an infinite set of possible scenarios. The algorithm can determine the robust values of the first-stage decision variables when the only information available to decision makers at the time of making the first stage decisions are a real compact interval containing possible values for each uncertain parameter with unknown probability distribution. The algorithm sequentially solves and updates a relaxation problem using decomposition algorithms, priority based methods, approximation algorithms, and bi-level optimizations until both feasibility and optimality conditions of the overall problem are satisfied. We significantly reduced the number of times the Bi-level 2 model had to be solved using priority methods to classify quality of solutions. Solving the Bi-level 2 method in the previous algorithm was a bottleneck with respect to computationally efficiency. Thus our results show that our algorithm provides an optimal solution with a significant improvement in computation time for the large-scale robust optimization problem.

We also prove that the algorithm terminates in a finite number of steps. The theoretical contribution of this research is that we generated new and improved approaches to generate Min-Max regret robust first-stage solution when the uncertain parameter independently takes its value from a real compact interval. We also came up with a practical approach for deviation robust decision making under uncertainty.

5.2 Applications of the proposed algorithm

The algorithm is very useful for facility location problems such as locating hospitals, distribution centers, manufacturing plants, where the first stage long term decision has to be made well in advance of the second stage decisions. In addition the concepts can also be applied to problems that involve human life such as evacuation during disaster scenarios. In such cases, since we are dealing with human lives, we cannot be satisfied with solutions that perform better on the average. Hence the deviation robust algorithm can be used to plan evacuation meeting points and also the infrastructure.

5.3 Future Research

Even though computation time was reduced significantly we believe the algorithm can be further improved by exploring the following methods. In practical applications the problem set can get very large. We can explore other meta-heuristic approximation methods instead of the Full Factorial Method as the Full Factorial method still takes a significant amount of time. In addition generalized Benders Decomposition method could be used to further improve the solution time of the bi-level programming problems.

The future direction of the research should be to explicitly explore the procedures suggested above in order to further improve the performance of our algorithm. The approach can also be extended to using the Relative Robust criterion to evaluate overall performance and compare it to the Deviation Robust method.

REFERENCES

Ahmed,S., Sahinidis,N. (1998), Robust Process Planning Under Uncertainty, *Industrial Engineering and Chemical Research*, vol. 37, 1883-1892.

Aiyoshi,E., Shimuzu,K. (1984), A Solution Method for the Static Constrained Stackelberg Problem via Penalty Method, *IEEE Transaction Automatic Control,* Vac-29, no.12, pp 1111-1114.

Anandalingam,G., Mathieu,R., Pittard,L., and Sinha,N. (1983). Artificial intelligence based approaches for solving hierarchical optimization problems. In Sharda,R., Golden,B., Wasil,E., Balci,O. and Stewart,W. editors, *Impacts of Recent Computer Advances on Operations Research*, pp 289–301.

Assavapokee,T., (2004), Semi Continuous Robust Approach for Strategic Infrastructure planning of Reverse Production Systems. Doctoral thesis submitted to the Faculty of the Department of Systems and Industrial Engineering at Georgia Institute of Technology.

Assavapokee,T., Realff,M.J., Ammons,J.C. and Hong,I. (2008a). A Scenario Relaxation Algorithm for Finite Scenario Based Min-Max Regret and Min-Max Relative Regret Robust Optimization. *Computers and Operations Research* vol. 35, no. 6, June 2008.

Assavapokee,T., Realff,M.J. and Ammons,J.C. (2008b). Min-Max Regret Robust Optimization Approach on Interval Data Uncertainty. *Journal of Optimization Theory and Applications*, vol. 136, no. 3, March 2008.

Assavapokee,T., Realff,M.J. and Ammons,J.C (2007a). A Min-Max Regret Robust Optimization Approach for Large Scale Full Factorial Scenario Design of Data Uncertainty. Under Review.

Assavapokee,T., Realff,M.J. and Ammons,J.C. (2007b). A Relative Robust Optimization Approach for Full Factorial Scenario Design of Data Uncertainty and Ambiguity. Under Review.

Averbakh,I. (2001), On the Complexity of a Class of Combinatorial Optimization Problems with Uncertainty. *Math. Prog.*, vol. 90, pp 263-272.

Bai,D., Carpenter,T. and Mulvey,J. (1997), Making a Case for Robust Optimization Models. *Management Science.* vol. 43/7, pp 895-907.

Bard,J.F.(1991). Some Properties of Bi-level Programming Problem. *Journal of Optimization Theory and Application*, vol 68,no.2, pp 371-378

Bard,J.F. (1999). Practical Bi-level Optimization: Algorithms and Applications. Springer-Verlag.

Bard,J.F., Moore,J.T. (1990). A Branch and Bound Algorithm for the Bi-level Programming Problem. *SIAM Journal of Scientific and Statistical Computing,* vol 11, no.2, pp 281-292.

Baumol,W.J. and Fabian,T. (1964). Decomposition, pricing for decentralization and external economies. *Management Science,* vol 11, pp 1-32.

Benders,J.F. (1962), Partitioning Procedures for Solving Mixed Variables Programming Problems. *Numerische Mathematik,*vol 4, pp 238-252

Ben-Tal,A.and Nemirovski,A. (1998).Robust Convex Optimization. *Mathematics of Operations Research.* vol 23, no.4, pp 769-805

Ben-Tal,A. and Nemirovski,A. (1999). Robust solutions to uncertain programs. *Operations Research Letters,* vol 25, pp 1-13.

Ben-Tal,A. and Nemirovski,A. (2000).Robust Solutions of Linear Programming Problems contaminated with Uncertain Data. *Mathematical Programming,* vol 88, pp 411-424

Bertsimas,D. and Sim,M. (2003). Robust Discrete Optimization and Network Flows. *Mathematical Programming Series B,* vol 98, pp 49-71.

Bertsimas,D. and Sim,M. (2004). The Price of Robustness. *Operations Research,* vol 52, no.1, pp 35-53.

Bjorndal,M. and Jornsten,K. (2005). The Deregulated Electricity Market viewed as a Bi-level Programming Problem. *Journal of Global Optimization.* vol 33, pp 465-475.

Bok,J., Lee,H. and Park,S. (1998). Robust Investment Model for Long-Range Capacity Expansion of Chemical Processing Networks under Uncertain Demand Forecast Scenarios. *Computers and Chemical Engineering*, vol. 22, no.7, pp 1037-1049.

Butler,R.J. (2003). Supply Chain Design for New Product, Ph.D. Thesis at Georgia Institute of Technology.

Candler,W. and Townsley,R. (1982). A Linear Two-Level Programming Problem. *Computers and Operations Research*,vol 9, no.1, pp 59-76

Cao,D., and Chen,M. (2006). Capacitated plant selection in a decentralized manufacturing environment: A bi-level optimization approach. *European Journal of Operational Research*, vol 169, pp 97-110.

Dantzig,G.B. and Wolfe,P. (1960). Decomposition Principle for Linear Programs. *Operations Research*, vol 8, no.1, pp 101-111

Dempe, S., Kalashnikov.V. and Rios-Mercado,R. (2005). *European Journal of Operational Research*, vol 166, pp 469-488.

Dempe,S. (1996b). Discrete bi-level optimization problems. *Technical Report 12*, Universitat Leipzig, Wirtschaftswissenschaftliche Fakultat. http://www.mathe.tu-freiberg.de/~dempe.

Edmunds,T.A. and Bard,J.F. (1991) Algorithms for Nonlinear Bi-level Mathematical Programs. *IEEE Transactions on Systems, Man and Cybernetics*, vol 21,no.1, pp 83-89.

Edmunds,T.A. and Bard,J.F.(1992). An algorithm for the mixed-integer nonlinear bi-level programming problem. *Annals of Operations Research*, vol 34, pp 149–162.

Falk,J.E. (1967). Lagrange Multipliers and Nonlinear Programming. *Journal of Mathematical Analysis*, vol 19, pp 141-159.

Friesz,T. L., Cho,H.-J., Mehta,N. J.,Tobin,R. L., and Anandalingam,G. (1992). A simulated annealing approach to the network design problem with variational inequality constraints. *Transportation Science*, vol 26, pp 18–26.

Friesz,T. L., Anandalingam,G., Mehta,N. J., Nam,K., Shah,S.J., and Tobin,R. L. (1993). The multi objective equilibrium network design problem revisited: A simulated annealing approach. *European Journal of Operational Research*, vol 65, pp 44–57.

Gendreau,M., Marcotte,P., and Savard,G. (1996). A hybrid tabu-ascent algorithm for the linear bi-level programming problem. *Journal of Global Optimization*, vol 8, pp 217–233.

Geoffrion,A.M. (1972). Generalized Benders Decomposition. *Journal of Optimization Theory and Application,* vol 10, no.4, pp 237-260.

Geoffrion,A.M. and Graves,G.W. (1974). Multi commodity Distribution System Design by Benders Decomposition. *Management Science*, vol 20,no.5, pp 822-844

Goetschalckx,M., Ahmed,S., Shapiro,A. and Santoso,T. (2001), Designing Flexible and Robust Supply Chain, IEPM Quebec.

Gutierrez,G.J., Kouvelis,P. and Kurawarwala,A.A. (1996). A robustness approach to uncapacitated network design problems. *European Journal of Operations Research.* v94, pp 362-376

Hodder,J. and Dincer,M.C. (1986). A Multifactor Model for International Plant Location and Financing Under Uncertainty. *Computers and Operations Research*, vol. 13, no. 5, pp 601-609.

Jeroslow,R.G. (1985). The Polynomial Hierarchy and a Simple Model for Competitive Analysis. *Mathematical Programming,* vol 32, pp146-164.

Judice,J.J. and Faustino,A.M. (1992). A Sequential LCP Method for Bi-level Linear Programming. *Annals of Operations Research* ,vol 34, no.1, pp 89-106.

Kouvelis,P. and Yu,G. (1996). *Robust Discrete Optimization and its Applications, Vol 14.* Kluwer Academic Publishers.

Magnanti,T.L. and Wong,R.T. (1981).Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria. *Operations Research,* vol 29,no. 3, pp 464-484.

Magnanti,T., Mireault,P. and Wong,R.T. (1986). Tailoring Benders decomposition for uncapacitated network design. *Mathematical Programming Studies,* vol 26, pp 112-154.

Marcotte, P. (1986). Network design problem with congestion effects: a case of bi-level programming. *Mathematical Programming,* vol 34, pp142–162.

Marcotte, P., Savard, G., and Zhu, D. L. (2001). A trust region algorithm for nonlinear bi-level programming. *Operations Research Letters,* vol 29, pp 171–179.

Montemanni,R. and Gambardella,L.M. (2002).An algorithm for the relative robust shortest path problem with interval data. *Technical Report IDSIA-05-02*

Moore,J. and Bard,J.F. (1990). The mixed integer linear bi-level programming problem. *Operations Research,* vol 38, pp 911–921.

Mulvey, J., Vanderbei,R. and Zenios,S. (1995). Robust Optimization of Large-Scale Systems. *Operation Research*, vol. 43, pp 264-281.

Newton J, D. (2000). A Robust Approach for Planning the Strategic Infrastructure of Reverse Production System. Ph.D. Thesis at Georgia Institute of Technology.

Randazzo,C.D. and Luna,H.P.L.(2001) A comparison of optimal methods for local access uncapacitated network design. *Annals of Operations Research,* vol 106, pp 263-286.

Santoso,T., Ahmed,S., Goetschalckx,M. and Shapiro,A. (2005). A Stochastic Programming Approach for Supply Chain Network Design Under Uncertainty. *European Journal of Operational Research,* vol167, pp 96-115.

Scholtes,S. and Stohr,M. (1999). Exact penalization of mathematical programs with equilibrium constraints. *SIAM Journal on Control and Optimization,* vol 37, pp 617–652.

Shimizu,K., Lu,M. (1995), A Global Optimization Method for the Stackelberg Problem with Convex Functions via Problem Transformations and Concave Programming. *IEEE Transaction System, Man and Cybernetics,* vol 25, no.12, pp 1635-1640.

Soyster,A.L.(1973). Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research,* vol 21, pp 1154-1157.

Taha,H.A.(1997). *Operations Research-An Introduction,*6th edition, Prentice Hall.

Terlaky,T (1996). *Interior Point Methods in Mathematical Programming,* Kluwer Academic Publisher.

Thirwani,D. and Arora,S.R. (1997). An algorithm for the integer linear fractional bi-level programming problem. *Optimization,* vol 39, pp 53–67.

Von Stackelberg,H. (1943). Grundzuge der Theoretischen Volkswirtschaftslehre Stuttgart, Berlin: W. Kohlhammer.

Von Stackelberg, H. (1952). *The Theory of the Market Economy.* Translated from the German and with an introduction. by Alan T. Peacock. New York, Oxford University Press

Waypuparb,P. (2005). Reverse Logistics Network Design for Electronic Products in the State of Texas. *Masters Thesis.* Department of Industrial Engineering, University of Houston

Wen,U. and Yang,Y. (1990). Algorithms for solving the mixed integer two-level linear programming problem. *Computers and Operations Research,* vol 17, pp 133–142.

Yu, C., H. Li (2000), "A Robust Optimization Model for Stochastic Logistics Problems. *International Journal of Production Economics*, vol. 64, pp 385-397.