A MIN-MAX REGRET ROBUST OPTIMIZATION APPROACH FOR LARGE SCALE FULL FACTORIAL SCENARIO DESIGN OF DATA UNCERTAINTY

Tiravat Assavapokee
Department of Industrial Engineering, University of Houston,
Houston, Texas 77204-4008, USA, tiravat.assavapokee@mail.uh.edu

Matthew J. Realff
Department of Chemical and Biomolecular Engineering, Georgia Institute of Technology,
Atlanta, Georgia, 30332-0100, USA, matthew.realff@chbe.gatech.edu

Jane C. Ammons
Department of Industrial and Systems Engineering, Georgia Institute of Technology,
Atlanta, Georgia, 30332-0100, USA, jane.ammons@isye.gatech.edu

This paper presents a three-stage optimization algorithm for solving two-stage robust decision making problems under uncertainty with min-max regret objective. The structure of the first stage problem is a general mixed integer (binary) linear programming model with a specific model of uncertainty that can occur in any of the parameters, and the second stage problem is a linear programming model. Each uncertain parameter can take its value from a finite set of real numbers with unknown probability distribution independently of other parameters' settings. This structure of parametric uncertainty is referred to in this paper as the full-factorial scenario design of data uncertainty. The proposed algorithm is shown to be efficient for solving large-scale min-max regret robust optimization problems with this structure. The algorithm coordinates three mathematical programming formulations to solve the overall optimization problem. The main contributions of this paper are the theoretical development of the three-stage optimization algorithm, and improving its computational performance through model transformation, decomposition, and pre-processing techniques based on analysis of the problem structure. The proposed algorithm is applied to solve a number of robust facility location problems under this structure of parametric uncertainty. All results illustrate significant improvement in computation time of the proposed algorithm over existing approaches.

Keywords: Robust optimization, Scenario based decision making, Decision making under uncertainty, Min-max regret robust approach, Full-factorial scenario design

1. Introduction

In this paper we address the two-stage decision making problem under uncertainty, where the uncertainty can appear in the values of any parameters of a general mixed integer linear programming formulation.

$$\begin{aligned} \max_{\vec{x},\vec{y}} \quad & \vec{c}^T\vec{x} + \vec{q}^T\vec{y} \\ \text{s.t.} \quad & \boldsymbol{W_1}\vec{y} \le \vec{h}_1 + \boldsymbol{T_1}\vec{x} \\ & \boldsymbol{W_2}\vec{y} = \vec{h}_2 + \boldsymbol{T_2}\vec{x} \\ & \vec{x} \in \{0,1\}^{|\vec{x}|}, \quad \vec{y} \ge \vec{0} \end{aligned}$$

In this paper, lower case letters with vector cap such as $\vec{x}$ represent vectors and the notation x_i represents the i^{th} element of the vector $\vec{x}$. The corresponding bold upper case letters such as $\boldsymbol{W}$ denote matrices and the notation W_{ij} represents the $(i, j)^{\text{th}}$ element of the matrix $\boldsymbol{W}$.

In the formulation, the vector $\vec{x}$ represents the first-stage binary decisions that are made before the realization of uncertainty and the vector $\vec{y}$ represents the second-stage continuous decisions that are made after the uncertainty is resolved. It is worth emphasizing that many practical two-stage decision problems can often be represented by this mathematical formulation. For example in a facility location problem under uncertainty in a given network $G = (N, A)$, the first-stage decision $\vec{x} \in \{0,1\}^{|N|}$ represents the decision of whether or not a facility will be located at each node in the network *G*. These facility location decisions are considered as long-term decisions and have to be made before the realization of uncertain parameters in the model. Once all values of uncertain parameters in the model are realized, the second-stage decision $\vec{y} \ge \vec{0}$ can be made. These second-stage decisions represent the flow of each material transported on each arc in the network *G*.

In this paper, we consider the problem such that each uncertain parameter in the model is restricted to independently take its value from a finite set of real numbers with unknown probability distribution. This captures situations where decision makers have to make the long term decision under uncertainty when there is very limited or no historical information about the problem and the only available information is the possible value of each uncertain parameter based on expert opinion. For example, expert opinion may give the decision makers three possible values for a given uncertain parameter based on optimistic, average, and pessimistic estimations. Under this situation, decision makers cannot search for the first-stage decisions with the best long run average performance, because there is a lack of knowledge about the probability distribution of the problem parameters. Instead, decision makers are assumed to require first-stage decisions that perform well across all possible input scenarios: robust first-stage decisions using a deviation robustness definition (min-max regret robust solution) defined by Kouvelis and Yu (1997).

Two-stage min-max regret robust optimization addresses optimization problems where some of the model parameters are uncertain at the time of making the first stage decisions. The criterion for the first stage decisions is to minimize the maximum regret between the optimal objective function value under perfect information and the resulting objective function value under the robust decisions over all possible realizations of the uncertain parameters (scenarios) in the model. The work of Kouvelis and Yu (1997) summarizes the state-of-the-art in min-max regret optimization up to 1997 and provides a comprehensive discussion of the motivation for the min-max regret approach and various aspects of applying it in practice. Ben-Tal and Nemirovski (1998, 1999, 2000) address 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. Mausser and Laguna (1998, 1999a, 1999b) presents the mixed integer linear programming formulation to finding the maximum regret scenario for a given candidate robust solution of one-stage linear programming problems under interval data uncertainty for objective function coefficients. They also develop an iterative algorithm for finding the robust solution of one-stage linear programming problems under relative robust criterion and a heuristic algorithm under the min-max absolute regret criterion with the similar type of uncertainty. Averbakh (2000, 2001) shows 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 definition. Bertsimas and Sim (2003, 2004) propose an approach to address data uncertainty for discrete optimization and network flow problems that allows the degree of conservatism of the solution (min-max/max-min objective) to be controlled. They show that the robust counterpart of an *NP*-hard α-approximable 0-1 discrete optimization problem remains α-approximable. They also 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. Assavapokee et al. (2007a) present a scenario relaxation algorithm for solving scenario-based min-max regret and min-max relative regret robust optimization problems for the mixed integer linear programming

(MILP) formulation. Assavapokee et al. (2007b) present a min-max regret robust optimization algorithm for two-stage decision making problems under interval data uncertainty in parameter vectors $\vec{c}, \vec{h}_1, \vec{h}_2$ and parameter matrixes $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ of the MILP formulation. They also demonstrate a counter-example illustrating the insufficiency of the robust solution obtained by only considering scenarios generated by all combinations of upper and lower bounds of each uncertain parameter for the interval data uncertainty case. This result is also true for the full factorial scenario design of data uncertainty case. As of our best knowledge, there exists no algorithm for solving two-stage min-max regret robust decision making problems under interval data uncertainty in parameter vector $\vec{q}$ and parameter matrixes $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ of the general MILP formulation. Because the proposed algorithm in this paper can solve the similar robust two-stage problems under full factorial scenario design of data uncertainty in any parameter of the general MILP formulation, the proposed algorithm can also be used as one alternative to approximately solve the deviation robust two-stage problems under interval data uncertainty in parameter vector $\vec{q}$ and parameter matrixes $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ of the general MILP formulation by replacing the possible range of each uncertain parameter with a finite set of real values.

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 linear programming model. The extensive form model is explained in detail in section 2. 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. Under full-factorial scenario design, the number of scenarios grows exponentially with the number of uncertain parameters and the number of possible values for each uncertain parameter. For example, a problem with 20 uncertain parameters each with 3 possible values has over 3.4 billion scenarios. Solving the extensive form model directly obviously is not the efficient way for solving this type of robust decision problems even with the use of Benders' decomposition technique (J.F. Benders, 1962). For example, if the problem contains 3.4 billion scenarios and a sub-problem of Benders' decomposition can be solved within 0.01 second, the algorithm would require approximately one year of computation time per iteration to generate a set of required cuts.

Because of the failure of the extensive form model and the Benders' decomposition algorithm for solving a large scale problem of this type, the three-stage optimization algorithm is proposed in this paper for solving this type of robust optimization problems under min-max regret objective. The algorithm is designed explicitly to efficiently handle a combinatorial sized set of possible scenarios. 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 detail in section 2. The algorithm is proven to terminate at an optimal min-max regret robust solution if one exists in a finite number of iterations. Pre-processing procedures, problem transformation steps, and decomposition techniques are also derived to improve the computational tractability of the algorithm. In the following section, we present the theoretical methodology of the proposed algorithm. The performance of the proposed algorithm is demonstrated through a number of facility location problems in section 3. All results illustrate impressive computational performance of the algorithm on examples of practical size.

2. Methodology

This section begins by reviewing key concepts of scenario based min-max regret robust optimization. The methodology of the proposed algorithm is then summarized and explained in detail, and each of its three stages is specified. The section concludes with the proof that the proposed algorithm always terminates at the min-max regret robust optimal solution if one exists in a finite number of iterations.

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 $\vec{c}_\omega, \vec{q}_\omega, \vec{h}_{1\omega}, \vec{h}_{2\omega}$, $\boldsymbol{W}_{1\omega}, \boldsymbol{W}_{2\omega}, \boldsymbol{T}_{1\omega}$, and $\boldsymbol{T}_{2\omega}$ 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).

$$
\begin{aligned}
O_\omega^* = \max_{\vec{x}_\omega, \vec{y}_\omega} \quad & c_\omega^T \vec{x}_\omega + \vec{q}_\omega^T \vec{y}_\omega \\
\text{s.t.} \quad & \boldsymbol{W}_{\mathbf{1\omega}} \vec{y}_\omega - \boldsymbol{T}_{\mathbf{1\omega}} \vec{x}_\omega \le \vec{h}_{1\omega} \\
& \boldsymbol{W}_{\mathbf{2\omega}} \vec{y}_\omega - \boldsymbol{T}_{\mathbf{2\omega}} \vec{x}_\omega = \vec{h}_{2\omega} \\
& \vec{x}_\omega \in \{0,1\}^{|\vec{x}_\omega|} \text{ and } \vec{y}_\omega \ge \vec{0}
\end{aligned}
\tag{1}
$$

When parameters' uncertainty exists, the search for the min-max regret robust solution comprises finding binary choice decisions, $\vec{x}$, such that the function $\max_{\omega \in \bar{\Omega}} (O_\omega^* - Z_\omega^*(\vec{x}))$ is minimized where $Z_\omega^*(\vec{x}) = \max_{\vec{y}_\omega \ge \vec{0}} \left\{ \vec{q}_\omega^T \vec{y}_\omega \mid \boldsymbol{W}_{\mathbf{1\omega}} \vec{y}_\omega \le \vec{h}_{1\omega} + \boldsymbol{T}_{\mathbf{1\omega}} \vec{x},\ \boldsymbol{W}_{\mathbf{2\omega}} \vec{y}_\omega = \vec{h}_{2\omega} + \boldsymbol{T}_{\mathbf{2\omega}} \vec{x} \right\} + \vec{c}_\omega^T \vec{x}$. It is also worth mentioning that the value of $O_\omega^* - Z_\omega^*(\vec{x})$ is nonnegative for all scenarios $\omega \in \bar{\Omega}$.

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

$$
\begin{aligned}
\min_{\delta, \vec{x}, \vec{y}_\omega} \quad & \delta \\
\text{s.t.} \quad & \left.
\begin{aligned}
& \delta \ge O_\omega^* - \vec{q}_\omega^T \vec{y}_\omega - \vec{c}_\omega^T \vec{x} \\
& \boldsymbol{W}_{\mathbf{1\omega}} \vec{y}_\omega - \boldsymbol{T}_{\mathbf{1\omega}} \vec{x} \le \vec{h}_{1\omega} \\
& \boldsymbol{W}_{\mathbf{2\omega}} \vec{y}_\omega - \boldsymbol{T}_{\mathbf{2\omega}} \vec{x} = \vec{h}_{2\omega} \\
& \vec{y}_\omega \ge \vec{0}
\end{aligned}
\right\} \quad \forall \omega \in \bar{\Omega} \\
& \vec{x} \in \{0,1\}^{|\vec{x}|}
\end{aligned}
\tag{2}
$$

This model (2) is 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 the first stage decision variables $\vec{x}$. Unfortunately, the size of the extensive form model can become unmanageably large as does the required computation time to find the optimal setting of $\vec{x}$. 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 these approaches is proposed in the following subsection. A key insight of this algorithm is to recognize that, for a finitely large set of all possible scenarios, it is possible to identify a smaller subset of important scenarios that actually need to be considered as part of the iteration scheme to solve the overall optimization problem with the use of bi-level programming.

Min-Max Regret Robust Algorithm for Full-Factorial Scenario Design

We start this section by defining some additional notations which are extensively used in this section of the paper. The uncertain parameters in the model (1) can be classified into eight major types. These uncertain parameters can be combined into a random vector ξ = $(\vec{c}, \vec{q}, \vec{h}_1, \vec{h}_2, \boldsymbol{T_1}, \boldsymbol{T_2}, \boldsymbol{W_1}, \boldsymbol{W_2})$. In this paper, we assume that each component of ξ can independently take its values from a finite set of real numbers with unknown probability distribution. This means that for any element p of the vector ξ, p can take any value from the set of $\{p_{(1)}, p_{(2)}, ..., p_{(\bar{p})}\}$ such that $p_{(1)} < p_{(2)} < ... < p_{(\bar{p})}$ where the notation $\bar{p}$ denotes the number of possible values for the parameter p. For simplicity, notations $p^L = p_{(1)}$ and $p^U = p_{(\bar{p})}$ will be used to represent the lower bound and upper bound values of the parameter $p \in \xi$. 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. As described below, we propose a three-stage optimization algorithm for solving the min-max regret robust optimization problem under scenario set $\bar{\Omega}$ that utilizes the creative idea based on the following inequality where $\Omega \subseteq \bar{\Omega}$.

$$\Delta^U = \max_{\omega \in \bar{\Omega}} \{O^*_\omega - Z^*_\omega(\vec{x})\} \geq \min_{\vec{x}} \max_{\omega \in \bar{\Omega}} \{O^*_\omega - Z^*_\omega(\vec{x})\} \geq \min_{\vec{x}} \max_{\omega \in \Omega} \{O^*_\omega - Z^*_\omega(\vec{x})\} = \Delta^L$$

In the considered problem, we would like to solve the middle problem ($\min_{\vec{x}} \max_{\omega \in \bar{\Omega}} \{O^*_\omega - Z^*_\omega(\vec{x})\}$), which is intractable because $|\bar{\Omega}|$ is extremely large. Instead, we successively solve the left and right problems for Δ^U and Δ^L. The left problem is solved by utilizing a reformulation as a tractable Bi-level model. The right problem is solved by utilizing the Benders' decomposition based on the fact that $|\Omega|$ is relatively small compared to $|\bar{\Omega}|$. The proposed algorithm proceeds as follows.

Three-Stage Algorithm

***Step* 0:** (Initialization) Choose a subset $\Omega \subseteq \bar{\Omega}$ and set $\Delta^U = \infty$, and $\Delta^L = 0$. Let $\vec{x}_{opt}$ denote the incumbent solution. Determine the value of $\varepsilon \geq 0$, which is a pre-specified tolerance.

***Step* 1:** (Solving the Relaxation Problem and Optimality Check) 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 . By using the information on $O^*_\omega\ \forall\omega\in\Omega$, Apply the Benders' decomposition algorithm explained in detail in Section 2.1 to solve the smaller version of the model (2) by considering only the scenario set Ω instead of $\bar{\Omega}$. This smaller version of the model (2) is referred to as the relaxed model (2) in this paper. If the relaxed model (2) is infeasible, the algorithm is terminated with the confirmation that no robust solution exists for the problem. Otherwise, set $\vec{x}_\Omega = \vec{x}^*$ which is an optimal solution from the relaxed model (2) and set $\Delta^L = \max_{\omega\in\Omega}\{O^*_\omega - Z^*_\omega(\vec{x}^*)\}$ which is the optimal objective function value of the relaxed model (2). If $\{\Delta^U - \Delta^L\} \leq \varepsilon$, $\vec{x}_{opt}$ is the globally ε-optimal robust solution and the algorithm is terminated. Otherwise the algorithm proceeds to Step 2.

***Step* 2:** (Feasibility Check) Solve the Bi-level-1 model described in detail in Section 2.2 by using the $\vec{x}_\Omega$ information from Step 1. If the optimal objective function value of the Bi-level-1 model is nonnegative (feasible case), proceed to Step 3. Otherwise (infeasible case), $\Omega \leftarrow \Omega \cup \{\omega^*_1\}$ where ω^*_1 is the infeasible scenario for $\vec{x}_\Omega$ generated by the Bi-level-1 model in this iteration and return to Step 1.

***Step* 3:** (Generate the Scenario with Maximum Regret Value for $\vec{x}_\Omega$ and Optimality Check) Solve the Bi-level-2 model specified in detail in Section 2.3 by using the $\vec{x}_\Omega$ information from Step 2. Let $\omega^*_2 \in \arg\max_{\omega\in\bar{\Omega}}\{O^*_\omega - Z^*_\omega(\vec{x}_\Omega)\}$ and $\Delta^{U*} = \max_{\omega\in\bar{\Omega}}\{O^*_\omega - Z^*_\omega(\vec{x}_\Omega)\}$ generated by the Bi-level-2 model respectively in this iteration.

If $\Delta^{U*} < \Delta^U$, then set $\vec{x}_{opt} = \vec{x}_\Omega$ and set $\Delta^U = \Delta^{U*}$.

If $\{\Delta^U - \Delta^L\} \leq \varepsilon$, $\vec{x}_{opt}$ is the globally ε-optimal robust solution and the algorithm is terminated. Otherwise, $\Omega \leftarrow \Omega \cup \{\omega^*_2\}$ and return to Step 1.

We define the algorithm Steps 1, 2, and 3 as the first, second, and third stages of the algorithm respectively. Figure 1 illustrates a schematic structure of this algorithm. Each of the three stages of the algorithm is detailed in the following subsections.

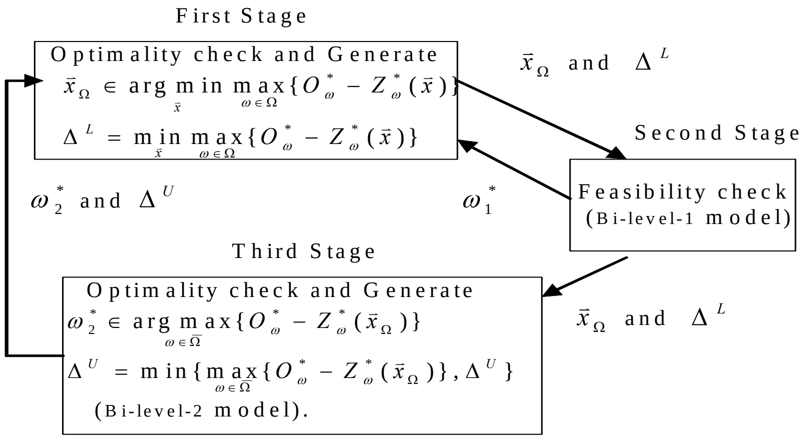

Figure 1: Schematic Structure of the Algorithm.

2.1. The First Stage Algorithm

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 paper, we recommend solving the relaxed model (2) by utilizing the Benders' decomposition techniques as follow. The relaxed model (2) has the following structure.

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

This model can also be rewritten as $\min_{\vec{x}\in\{0,1\}^{|\vec{x}|}} f(\vec{x})$ where $f(\vec{x}) = \max_{\omega\in\Omega}(O_\omega^* - Q_\omega(\vec{x}) - c_\omega^T \vec{x})$ and

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

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: $f(\vec{x}) = \max_{\omega\in\Omega}(O^*_\omega - Q_\omega(\vec{x}) - c^T_\omega \vec{x})$ is a convex function on $\vec{x}$ because of the following reasons. (1) $Q_\omega(\vec{x})$ and $c^T_\omega \vec{x}$ 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:** $\left(-\vec{c}^{\,T}_{\omega(i)} - (\vec{\pi}^*_{1,\omega(i),\vec{x}(i)})^T \boldsymbol{T_{1\omega(i)}} - (\vec{\pi}^*_{2,\omega(i),\vec{x}(i)})^T \boldsymbol{T_{2\omega(i)}}\right)^T \in \partial f(\vec{x}(i))$ where $\omega(i) \in \arg\max_{\omega\in\Omega}\{O^*_\omega - \vec{c}^{\,T}_\omega \vec{x}(i) - Q_\omega(\vec{x}(i))\}$, $\partial f(\vec{x}(i))$ 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 $Q_{\omega(i)}(\vec{x}) \le (\vec{\pi}^*_{1,\omega(i),\vec{x}(i)})^T(\vec{h}_{1\omega(i)} + \boldsymbol{T_{1\omega(i)}}\vec{x}) + (\vec{\pi}^*_{2,\omega(i),\vec{x}(i)})^T(\vec{h}_{2\omega(i)} + \boldsymbol{T_{2\omega(i)}}\vec{x})$ for arbitrary $\vec{x}$.

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

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

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

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

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

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

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* = –∞ and *ub* = +∞ 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

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

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:

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

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 $\omega(k) \in \arg\max_{\omega\in\Omega}\{O^*_\omega - \vec{c}_\omega^T \vec{x}(k) - Q_\omega(\vec{x}(k))\}$. 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 $ub - lb \le \lambda$, where $\lambda \ge 0$ 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 $\omega(k) \in \arg\max_{\omega\in\Omega}\{O^*_\omega - \vec{c}_\omega^T \vec{x}(k) - Q_\omega(\vec{x}(k))\}$, 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 $\vec{a}_k^T = -\left(\vec{c}^T_{\omega(k)} + (\vec{\pi}^*_{1,\omega(k),\vec{x}(k)})^T \boldsymbol{T}_{\mathbf{1\omega(k)}} + (\vec{\pi}^*_{2,\omega(k),\vec{x}(k)})^T \boldsymbol{T}_{\mathbf{2\omega(k)}}\right)$, and $b_k = -\vec{a}_k^T \vec{x}(k) + f(\vec{x}(k))$, 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.

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

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

2.2. The Second Stage Algorithm

The main purpose of the second stage algorithm is to identify a scenario $\omega_1^* \in \bar{\Omega}$ which admits no feasible solution to $Z^*_\omega(\vec{x}_\Omega)$ 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.

One can find a model parameters' setting or a scenario $\omega_1^* \in \bar{\Omega}$ which admits no feasible solution to $Z^*_\omega(\vec{x}_\Omega)$ for $\omega = \omega_1^*$ by solving the following bi-level programming problem referred to as the Bi-level-1 model. The following model (3) demonstrates the general structure of the Bi-level-1 model.

$$
\begin{aligned}
&\min_{\xi} && \delta \\
&\text{s.t.} && \xi \in \Xi \\
& && \max_{\vec{y},\vec{s},\vec{s}_1,\vec{s}_2,\delta} \quad \delta \\
& && \text{s.t.} \quad \boldsymbol{W_1}\vec{y} + \vec{s} = \vec{h}_1 + \boldsymbol{T_1}\vec{x}_\Omega \\
& && \qquad \boldsymbol{W_2}\vec{y} + \vec{s}_1 = \vec{h}_2 + \boldsymbol{T_2}\vec{x}_\Omega \\
& && \qquad -\boldsymbol{W_2}\vec{y} + \vec{s}_2 = -\vec{h}_2 - \boldsymbol{T_2}\vec{x}_\Omega \\
& && \qquad \delta\vec{1} \le \vec{s}, \quad \delta\vec{1} \le \vec{s}_1, \quad \delta\vec{1} \le \vec{s}_2, \quad \vec{y} \ge \vec{0}
\end{aligned} \tag{3}
$$

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 (3), 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 (3) has a nonlinear bi-level structure with a set of constraints restricting the possible values of the decision vectors $\xi = (\vec{c}, \vec{q}, \vec{h}_1, \vec{h}_2, \boldsymbol{T_1}, \boldsymbol{T_2}, \boldsymbol{W_1}, \boldsymbol{W_2})$. Because the structure of the follower problem of the model (3) 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 (3), all elements in decision variable matrixes $\boldsymbol{T_1}, \boldsymbol{W_1}$ and vector $\vec{h}_1$ can be predetermined to either one of their bounds even before solving the model (3). For each element of the decision matrix $\boldsymbol{W_1}$ in the model (3), 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 $\vec{y} \ge \vec{0}$. Similarly, for each element of the decision vector $\vec{h}_1$ and matrix $\boldsymbol{T_1}$, the optimal setting of this decision variable in the model (3) is the lower bound of its possible values.

***Lemma* 3:** The model (3) has at least one optimal solution $\boldsymbol{T_1^*}, \vec{h}_1^*, \boldsymbol{W_1^*}, \boldsymbol{T_2^*}, \vec{h}_2^*,$ and $\boldsymbol{W_2^*}$ 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 $\boldsymbol{T_1}, \vec{h}_1,$ and $\boldsymbol{W_1}$ already takes its value from one of its bounds. We only need to prove this Lemma for each element of $\boldsymbol{T_2}, \vec{h}_2,$ and $\boldsymbol{W_2}$. Each of these variables $T_{2il}, h_{2i},$ and W_{2ij} appears in only two constrains in the model

(3): $\sum_j W_{2ij} y_j + s_{1i} = h_{2i} + \sum_l T_{2il} x_{\Omega l}$ 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 $\bar{y}$ which maximizes $\min\{s_{1i}, s_{2i}\}$ will also minimize $|s_{1i} - s_{2i}|/2$ and vise 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 $\min_{\bar{y} \in \chi(\bar{x}_\Omega)} |h_{2i} + \sum_l T_{2il} x_{\Omega l} - \sum_j W_{2ij} y_j|$

where $\chi(\bar{x}_\Omega) = \{\bar{y} \geq \bar{0} \,|\, \boldsymbol{W_1} \bar{y} \leq \bar{h}_1 + \boldsymbol{T_1} \bar{x}_\Omega,\ \boldsymbol{W_2} \bar{y} \leq \bar{h}_2 + \boldsymbol{T_2} \bar{x}_\Omega\}$. 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 (3). Even though there are six sets of follower's constraints in the model (3), only three sets of dual variables are required. Because of the structure of dual constraints of the follower problem in the model (3), 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 form 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 constraints (4), (5), and (6) 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.

$$\left.\begin{array}{l}
T_{2il} = T_{2il}^{L} + (T_{2il}^{U} - T_{2il}^{L}) biT_{2il} \\
T_{2il}^{L} w_{2i}^{+} \leq TW_{2il}^{+} \leq T_{2il}^{U} w_{2i}^{+} \\
T_{2il}^{L} w_{2i}^{-} \leq TW_{2il}^{-} \leq T_{2il}^{U} w_{2i}^{-} \\
TW_{2il}^{+} \geq T_{2il}^{U} w_{2i}^{+} - (|T_{2il}^{U}| + |T_{2il}^{L}|)(w_{2i}^{+U})(1 - biT_{2il}) \\
TW_{2il}^{+} \leq T_{2il}^{L} w_{2i}^{+} + (|T_{2il}^{U}| + |T_{2il}^{L}|)(w_{2i}^{+U})(biT_{2il}) \\
TW_{2il}^{-} \geq T_{2il}^{U} w_{2i}^{-} - (|T_{2il}^{U}| + |T_{2il}^{L}|)(w_{2i}^{-U})(1 - biT_{2il}) \\
TW_{2il}^{-} \leq T_{2il}^{L} w_{2i}^{-} + (|T_{2il}^{U}| + |T_{2il}^{L}|)(w_{2i}^{-U})(biT_{2il}) \\
\qquad biT_{2il} \in \{0,1\}
\end{array}\right\} \quad \forall i \in E,\ \forall l \tag{4}$$

$$\left.\begin{array}{l}
h_{2i} = h_{2i}^{L} + (h_{2i}^{U} - h_{2i}^{L}) bih_{2i} \\
h_{2i}^{L} w_{2i}^{+} \leq HW_{2i}^{+} \leq h_{2i}^{U} w_{2i}^{+} \\
h_{2i}^{L} w_{2i}^{-} \leq HW_{2i}^{-} \leq h_{2i}^{U} w_{2i}^{-} \\
HW_{2i}^{+} \geq h_{2i}^{U} w_{2i}^{+} - (|h_{2i}^{U}| + |h_{2i}^{L}|)(w_{2i}^{+U})(1 - bih_{2i}) \\
HW_{2i}^{+} \leq h_{2i}^{L} w_{2i}^{+} + (|h_{2i}^{U}| + |h_{2i}^{L}|)(w_{2i}^{+U})(bih_{2i}) \\
HW_{2i}^{-} \geq h_{2i}^{U} w_{2i}^{-} - (|h_{2i}^{U}| + |h_{2i}^{L}|)(w_{2i}^{-U})(1 - bih_{2i}) \\
HW_{2i}^{-} \leq h_{2i}^{L} w_{2i}^{-} + (|h_{2i}^{U}| + |h_{2i}^{L}|)(w_{2i}^{-U})(bih_{2i}) \\
\qquad bih_{2i} \in \{0,1\}
\end{array}\right\} \quad \forall i \in E \tag{5}$$

$$\left.\begin{array}{l}
W_{2ij} = W_{2ij}^{L} + (W_{2ij}^{U} - W_{2ij}^{L}) biW_{2ij} \\
W_{2ij}^{L} y_{j} \leq WY_{2ij} \leq W_{2ij}^{U} y_{j} \\
WY_{2ij} \geq W_{2ij}^{U} y_{j} - (|W_{2ij}^{U}| + |W_{2ij}^{L}|)(y_{j}^{U})(1 - biW_{2ij}) \\
WY_{2ij} \leq W_{2ij}^{L} y_{j} + (|W_{2ij}^{U}| + |W_{2ij}^{L}|)(y_{j}^{U})(biW_{2ij}) \\
W_{2ij}^{L} w_{2i}^{+} \leq WW_{2ij}^{+} \leq W_{2ij}^{U} w_{2i}^{+} \\
W_{2ij}^{L} w_{2i}^{-} \leq WW_{2ij}^{-} \leq W_{2ij}^{U} w_{2i}^{-} \\
WW_{2ij}^{+} \geq W_{2ij}^{U} w_{2i}^{+} - (|W_{2ij}^{U}| + |W_{2ij}^{L}|)(w_{2i}^{+U})(1 - biW_{2ij}) \\
WW_{2ij}^{+} \leq W_{2ij}^{L} w_{2i}^{+} + (|W_{2ij}^{U}| + |W_{2ij}^{L}|)(w_{2i}^{+U})(biW_{2ij}) \\
WW_{2ij}^{-} \geq W_{2ij}^{U} w_{2i}^{-} - (|W_{2ij}^{U}| + |W_{2ij}^{L}|)(w_{2i}^{-U})(1 - biW_{2ij}) \\
WW_{2ij}^{-} \leq W_{2ij}^{L} w_{2i}^{-} + (|W_{2ij}^{U}| + |W_{2ij}^{L}|)(w_{2i}^{-U})(biW_{2ij}) \\
\qquad biW_{2ij} \in \{0,1\}
\end{array}\right\} \quad \forall i \in E,\ \forall j \tag{6}$$

After applying pre-processing rules, the follower problem transformation, and the result from Lemma 3, the model (3) can be transformed from a bi-level nonlinear structure to a single-level mixed integer linear structure presented in the model (7). The table in the model (7) is used to identify some additional constraints and conditions for adding these constraints to the model (7). 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 (7) 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 third stage algorithm.

$$
\begin{aligned}
\min \quad & \delta \\
\text{s.t.} \quad & \\
& \sum_j W_{1ij}^U y_j + s_i = h_{1i}^L + \sum_l T_{1il}^L x_{\Omega l} && \forall i \in L \\
& \sum_j WY_{2ij} + s_{1i} = h_{2i} + \sum_l T_{2il} x_{\Omega l} && \forall i \in E \\
& -\sum_j WY_{2ij} + s_{2i} = -h_{2i} - \sum_l T_{2il} x_{\Omega l} && \forall i \in E \\
& \delta \le s_i \quad \forall i \in L, \qquad \delta \le s_{1i} \quad \forall i \in E, \qquad \delta \le s_{2i} && \forall i \in E \\
& \sum_{i \in L} W_{1ij}^U w_{1i} + \sum_{i \in E} \left(WW_{2ij}^+ - WW_{2ij}^- \right) \ge 0 && \forall j \\
& \sum_{i \in L} w_{1i} + \sum_{i \in E} \left(w_{2i}^+ + w_{2i}^- \right) = 1 \\
& \delta = \sum_{i \in L} \left(h_{1i}^L + \sum_l T_{1il}^L x_{\Omega l} \right) w_{1i} + \sum_{i \in E} \left(HW_{2i}^+ - HW_{2i}^- + \sum_l \left(TW_{2il}^+ - TW_{2il}^- \right) x_{\Omega l} \right) \\
& w_{1i} \ge 0 \ \forall i \in L, \quad w_{2i}^+ \ge 0 \ \ \forall i \in E, \quad w_{2i}^- \ge 0 \ \forall i \in E, \quad y_j \ge 0 \ \forall j
\end{aligned}
\tag{7}
$$

Condition to Add the Constraints	Constraint Reference	Constraint Index Set
Always	(4)	For all $i \in E$, For all l
Always	(5)	For all $i \in E$
Always	(6)	For all $i \in E$, For all j

2.3. The Third Stage Algorithm

The main purpose of the third stage is to identify a scenario $\omega_2^* \in \arg\max_{\omega \in \bar{\Omega}} \left\{ O_\omega^* - Z_\omega^*(\vec{x}_\Omega) \right\}$. The mathematical model utilized by this stage to perform this task is also a bi-level program referred to as the Bi-level-2 model. The leader's objective is to find the setting of the decision vector $\xi = (\vec{c}, \vec{q}, \vec{h}_1, \vec{h}_2, \boldsymbol{T_1}, \boldsymbol{T_2}, \boldsymbol{W_1}, \boldsymbol{W_2})$ and decision vector $(\vec{x}_1, \vec{y}_1)$ that result in the maximum regret value possible, $\max_{\omega \in \bar{\Omega}} \{ O_\omega^* - Z_\omega^*(\vec{x}_\Omega) \}$, for the candidate robust solution $\vec{x}_\Omega$. 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-2 model is represented in the following model (8).

The solution methodology for solving the model (8) 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.

$$
\begin{aligned}
\max_{\vec{x}_1 \in \{0,1\}^{|\vec{x}_1|}, \vec{y}_1 \geq \vec{0}, \xi} & \{ \vec{q}^{\,T} \vec{y}_1 + \vec{c}^{\,T} \vec{x}_1 - \vec{q}^{\,T} \vec{y}_2 - \vec{c}^{\,T} \vec{x}_\Omega) \} \\
\text{s.t.} \quad & \xi \in \Xi \\
& \boldsymbol{W_1} \vec{y}_1 \leq \vec{h}_1 + \boldsymbol{T_1} \vec{x}_1 \\
& \boldsymbol{W_2} \vec{y}_1 = \vec{h}_2 + \boldsymbol{T_2} \vec{x}_1 \\
& \max_{\vec{y}_2 \geq \vec{0}} \vec{q}^{\,T} \vec{y}_2 \\
& \text{s.t.} \quad \boldsymbol{W_1} \vec{y}_2 \leq \vec{h}_1 + \boldsymbol{T_1} \vec{x}_\Omega \\
& \qquad\; \boldsymbol{W_2} \vec{y}_2 = \vec{h}_2 + \boldsymbol{T_2} \vec{x}_\Omega
\end{aligned}
\tag{8}
$$

2.3.1. Parameter Pre-Processing Step

From the structure of the model (8), 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 (8). 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 (8) 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 $\boldsymbol{T_1}$

The elements of decision vector $\boldsymbol{T_1}$ 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 $\boldsymbol{T_1}$ is represented in the constraint of the model (8) as: $\sum_j W_{1ij} y_{1j} \leq h_{1i} + T_{1il} x_{1l} + \sum_{k \neq l} T_{1ik} x_{1k}$ and $\sum_j W_{1ij} y_{2j} \leq h_{1i} + T_{1il} x_{\Omega l} + \sum_{k \neq l} T_{1ik} x_{\Omega k}$. From any given value of $x_{\Omega l}$, the value of T_{1il} can be predetermined at $T^*_{1il} = T^U_{1il}$ 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 (9) where the new variable TX_{11il} replaces the nonlinear term $T_{1il}x_{1l}$ in the model (8). The insight of this set of constraints (9) is that if the value of x_{1l} is set to be zero by the model, the optimal setting of T_{1il} is T^L_{1il} and $TX_{11il} = 0$. Otherwise the optimal setting of T_{1il} can not be predetermined and $TX_{11il} = T_{1il}$.

$$\begin{aligned} & TX_{11il} - T_{1il} + T^L_{1il}(1 - x_{1l}) \le 0 \\ & -TX_{11il} + T_{1il} - T^U_{1il}(1 - x_{1l}) \le 0 \\ & T^L_{1il}x_{1l} \le TX_{11il} \le T^U_{1il}x_{1l} \\ & T^L_{1il} \le T_{1il} \le T^L_{1il} + x_{1l}(T^U_{1il} - T^L_{1il}) \end{aligned} \tag{9}$$

2.3.2. Problem Transformation Step

In order to solve the model (8) 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 (8) into a computationally efficient formulation. The following two subsections describe techniques and methodologies for performing these two tasks.

2.3.2.1. 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: $p = \sum_{i=1}^{\bar{p}} p_{(i)} bi_i$, $\sum_{i=1}^{\bar{p}} bi_i = 1,\ bi_i \ge 0\ \forall i = 1, \dots, \bar{p}$ and $\{bi_1, bi_2, \dots, bi_{\bar{p}}\}$ 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, $bi_i\ \ \forall i = 1, \dots, \bar{p}$, are defined as SOS1, there are only $\bar{p}$ branches required in the searching tree for these variables.

2.3.2.2. Final Transformation Steps for the Bi-level-2 Model

Because the structure of the follower problem in the model (8) 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 (10) illustrates the formulation of the model (8) 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 (10) 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.

$$
\begin{aligned}
&\max\{\sum_j q_j y_{1j} + \sum_l c_l x_{1l} - \sum_j q_j y_{2j} - \sum_l c_l x_{\Omega l}\} \\
&\text{s.t.}\quad \xi \in \Xi \\
&\qquad \sum_j W_{1ij} y_{1j} \le h_{1i} + \sum_l T_{1il} x_{1l} \quad \forall i \in L \\
&\qquad \sum_j W_{1ij} y_{2j} \le h_{1i} + \sum_l T_{1il} x_{\Omega l} \quad \forall i \in L \\
&\qquad \sum_j W_{2ij} y_{1j} = h_{2i} + \sum_l T_{2il} x_{1l} \quad \forall i \in E \\
&\qquad \sum_j W_{2ij} y_{2j} = h_{2i} + \sum_l T_{2il} x_{\Omega l} \quad \forall i \in E \\
&\qquad \sum_{i\in L} W_{1ij} w_{1i} + \sum_{i \in E} W_{2ij} w_{2i} \ge q_j \quad \forall j \\
&\qquad \sum_{i\in L}\left(h_{1i} + \sum_l T_{1il} x_{\Omega l}\right) w_{1i} + \sum_{i\in E}\left(h_{2i} + \sum_l T_{2il} x_{\Omega l}\right) w_{2i} = \sum_j q_j y_{2j} \\
&\qquad w_{1i} \ge 0 \ \ \forall i \in L, \quad y_{1j} \ge 0, y_{2j} \ge 0 \ \ \forall j, \quad x_{1l} \in \{0,1\} \ \ \forall l
\end{aligned}
\tag{10}
$$

Final Transformation Steps

Parameter $\vec{c}$: 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 (9) to replace the nonlinear term $T_{1il}x_{1l}$ in the model (10), then add a set of variables and constraints illustrated in (11) to replace part of the constraint $\xi \in \Xi$ for T_{1il} in the model (10), and finally add a variable TW_{1il} and a set of variables and constraints illustrated in (12) to replace the nonlinear term $T_{1il}w_{1i}$ in the model (10), where w_{1i}^U and w_{1i}^L represent the upper bound and the lower bound of dual variable w_{1i} respectively.

$$
T_{1il} = \sum_{s=1}^{\bar{T}_{1il}} T_{1il(s)} biT_{1il(s)}, \sum_{s=1}^{\bar{T}_{1il}} biT_{1il(s)} = 1, \ \ biT_{1il(s)} \ge 0 \ \forall s \in \{1,2,...,\bar{T}_{1il}\} \text{ and } \bigcup_{\forall s}\{biT_{1il(s)}\} \text{ is SOS1} \tag{11}
$$

$$
\left.
\begin{aligned}
&TW_{1il} = \sum_{s=1}^{\bar{T}_{1il}} T_{1il(s)} ZTW_{1il(s)} \\
&w_{1i}^L biT_{1il(s)} \le ZTW_{1il(s)} \le w_{1i}^U biT_{1il(s)},\ \ ZTW_{1il(s)} \le w_{1i} - w_{1i}^L(1-biT_{1il(s)}),\ \ ZTW_{1il(s)} \ge w_{1i} - w_{1i}^U(1-biT_{1il(s)}) \quad \forall s \in \{1,...,\bar{T}_{1il}\}
\end{aligned}
\right\} \tag{12}
$$

Parameter T_2: We first add a decision variable TX_{21il} and a set of constraints illustrated in (13) to replace the nonlinear term $T_{2il}x_{1l}$ in the model (10), then add a set of variables and constraints illustrated in (14) to replace part of the constraint $\xi \in \Xi$ for T_{2il} in the model (10), and finally add a variable TW_{2il} and a set of variables and constraints illustrated in (15) to replace the nonlinear term $T_{2il}w_{2i}$ in the model (10), where w_{2i}^U and w_{2i}^L represent the upper bound and the lower bound of variable w_{2i} respectively.

$$\left.\begin{array}{l} TX_{21il} = \sum_{s=1}^{\bar{T}_{2il}} T_{2il(s)} ZTX_{21il(s)} \\ 0 \le ZTX_{21il(s)} \le 1,\ \ ZTX_{21il(s)} \le biT_{2il(s)},\ \ ZTX_{21il(s)} \le x_{1i},\ \ ZTX_{21il(s)} \ge biT_{2il(s)} + x_{1i} - 1\ \ \forall s \in \{1,...,\bar{T}_{2il}\} \end{array}\right\} \quad (13)$$

$$T_{2il} = \sum_{s=1}^{\bar{T}_{2il}} T_{2il(s)} biT_{2il(s)},\ \ \sum_{s=1}^{\bar{T}_{2il}} biT_{2il(s)} = 1,\ \ biT_{2il(s)} \ge 0\ \forall s \in \{1,2,...,\bar{T}_{2il}\} \text{ and } \bigcup_{\forall s}\{biT_{2il(s)}\} \text{ is SOS1} \quad (14)$$

$$\left.\begin{array}{l} TW_{2il} = \sum_{s=1}^{\bar{T}_{2il}} T_{2il(s)} ZTW_{2il(s)} \\ w_{2i}^L biT_{2il(s)} \le ZTW_{2il(s)} \le w_{2i}^U biT_{2il(s)},\ \ ZTW_{2il(s)} \le w_{2i} - w_{2i}^L(1 - biT_{2il(s)}),\ \ ZTW_{2il(s)} \ge w_{2i} - w_{2i}^U(1 - biT_{2il(s)})\quad \forall s \in \{1,...,\bar{T}_{2il}\} \end{array}\right\} (15)$$

Parameter $\vec{h}_1$ and $\vec{h}_2$: We first add a set of variables and constraints illustrated in (16) and (17) to replace part of the constraint $\xi \in \Xi$ for h_{1i} and h_{2i} respectively in the model (10). We then add variables HW_{1i}, HW_{2i} and a set of variables and constraints in (18) and (19) to replace the nonlinear terms $h_{1i}w_{1i}$ and $h_{2i}w_{2i}$ respectively in the model (10).

$$h_{1i} = \sum_{s=1}^{\bar{h}_{1i}} h_{1i(s)} biH_{1i(s)},\ \ \sum_{s=1}^{\bar{h}_{1i}} biH_{1i(s)} = 1,\ \ biH_{1i(s)} \ge 0\ \forall s \in \{1,2,...,\bar{h}_{1i}\} \text{ and } \bigcup_{\forall s}\{biH_{1i(s)}\} \text{ is SOS1} \quad (16)$$

$$h_{2i} = \sum_{s=1}^{\bar{h}_{2i}} h_{2i(s)} biH_{2i(s)},\ \ \sum_{s=1}^{\bar{h}_{2i}} biH_{2i(s)} = 1,\ \ biH_{2i(s)} \ge 0\ \forall s \in \{1,2,...,\bar{h}_{2i}\} \text{ and } \bigcup_{\forall s}\{biH_{2i(s)}\} \text{ is SOS1} \quad (17)$$

$$\left.\begin{array}{l} HW_{1i} = \sum_{s=1}^{\bar{h}_{1i}} h_{1i(s)} ZHW_{1i(s)} \\ w_{1i}^L biH_{1i(s)} \le ZHW_{1i(s)} \le w_{1i}^U biH_{1i(s)},\ \ ZHW_{1i(s)} \le w_{1i} - w_{1i}^L(1 - biH_{1i(s)}),\ \ ZHW_{1i(s)} \ge w_{1i} - w_{1i}^U(1 - biH_{1i(s)})\quad \forall s \in \{1,...,\bar{h}_{1i}\} \end{array}\right\} (18)$$

$$\left.\begin{array}{l} HW_{2i} = \sum_{s=1}^{\bar{h}_{2i}} h_{2i(s)} ZHW_{2i(s)} \\ w_{2i}^L biH_{2i(s)} \le ZHW_{2i(s)} \le w_{2i}^U biH_{2i(s)},\ \ ZHW_{2i(s)} \le w_{2i} - w_{2i}^L(1 - biH_{2i(s)}),\ \ ZHW_{2i(s)} \ge w_{2i} - w_{2i}^U(1 - biH_{2i(s)})\quad \forall s \in \{1,...,\bar{h}_{2i}\} \end{array}\right\} (19)$$

Parameter $\bar{q}$: We first add a set of variables and constraints illustrated in (20) to replace part of the constraint $\xi \in \Xi$ for q_j in the model (10). We then add decision variables QY_{1j}, QY_{2j} and a set of variables and constraints in (21) to replace the nonlinear terms $q_j y_{1j}$ and $q_j y_{2j}$ respectively in the model (10) 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.

$$q_j = \sum_{s=1}^{\bar{q}_j} q_{j(s)} biQ_{j(s)}, \quad \sum_{s=1}^{\bar{q}_j} biQ_{j(s)} = 1, \quad biQ_{j(s)} \geq 0 \ \forall s \in \{1,2,...,\bar{q}_j\} \text{ and } \bigcup_{\forall s}\{biQ_{j(s)}\} \text{ is SOS1} \quad (20)$$

$$\left.\begin{array}{l}
QY_{1j} = \sum_{s=1}^{\bar{q}_j} q_{j(s)} ZQY_{1j(s)} \\
y_{1j}^L biQ_{j(s)} \leq ZQY_{1j(s)} \leq y_{1j}^U biQ_{j(s)}, \ \ ZQY_{1j(s)} \leq y_{1j} - y_{1j}^L(1-biQ_{j(s)}), \ \ ZQY_{1j(s)} \geq y_{1j} - y_{1j}^U(1-biQ_{j(s)}) \quad \forall s \in \{1,...,\bar{q}_j\} \\
QY_{2j} = \sum_{s=1}^{\bar{q}_j} q_{j(s)} ZQY_{2j(s)} \\
y_{2j}^L biQ_{j(s)} \leq ZQY_{2j(s)} \leq y_{2j}^U biQ_{j(s)}, \ \ ZQY_{2j(s)} \leq y_{2j} - y_{2j}^L(1-biQ_{j(s)}), \ \ ZQY_{2j(s)} \geq y_{2j} - y_{2j}^U(1-biQ_{j(s)}) \quad \forall s \in \{1,...,\bar{q}_j\}
\end{array}\right\} (21)$$

Parameter W_1 and W_2: We first add a set of variables and constraints illustrated in (22) and (23) to replace part of the constraint $\xi \in \Xi$ for W_{1ij} and W_{2ij} respectively in the model (10). We then add a set of variables and constraints illustrated in (24) and (25) 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 (10).

$$W_{1ij} = \sum_{s=1}^{\bar{W}_{1ij}} W_{1ij(s)} biW_{1ij(s)}, \quad \sum_{s=1}^{\bar{W}_{1ij}} biW_{1ij(s)} = 1, \quad biW_{1ij(s)} \geq 0 \ \forall s \in \{1,2,...,\bar{W}_{1ij}\} \text{ and } \bigcup_{\forall s}\{biW_{1ij(s)}\} \text{ is SOS1} \quad (22)$$

$$W_{2ij} = \sum_{s=1}^{\bar{W}_{2ij}} W_{2ij(s)} biW_{2ij(s)}, \quad \sum_{s=1}^{\bar{W}_{2ij}} biW_{2ij(s)} = 1, \quad biW_{2ij(s)} \geq 0 \ \forall s \in \{1,2,...,\bar{W}_{2ij}\} \text{ and } \bigcup_{\forall s}\{biW_{2ij(s)}\} \text{ is SOS1} \quad (23)$$

$$\left.\begin{array}{l}
WY_{11ij} = \sum_{s=1}^{\bar{W}_{1ij}} W_{1ij(s)} ZWY_{11ij(s)} \\
y_{1j}^L biW_{1ij(s)} \leq ZWY_{11ij(s)} \leq y_{1j}^U biW_{1ij(s)}, \ \ ZWY_{11ij(s)} \leq y_{1j} - y_{1j}^L(1-biW_{1ij(s)}), \ \ ZWY_{11ij(s)} \geq y_{1j} - y_{1j}^U(1-biW_{1ij(s)}) \quad \forall s \in \{1,...,\bar{W}_{1ij}\} \\
WY_{12ij} = \sum_{s=1}^{\bar{W}_{1ij}} W_{1ij(s)} ZWY_{12ij(s)} \\
y_{2j}^L biW_{1ij(s)} \leq ZWY_{12ij(s)} \leq y_{2j}^U biW_{1ij(s)}, \ \ ZWY_{12ij(s)} \leq y_{2j} - y_{2j}^L(1-biW_{1ij(s)}), \ \ ZWY_{12ij(s)} \geq y_{2j} - y_{2j}^U(1-biW_{1ij(s)}) \quad \forall s \in \{1,...,\bar{W}_{1ij}\} \\
WW_{1ij} = \sum_{s=1}^{\bar{W}_{1ij}} W_{1ij(s)} ZWW_{1ij(s)} \\
w_{1i}^L biW_{1ij(s)} \leq ZWW_{1ij(s)} \leq w_{1i}^U biW_{1ij(s)}, \ \ ZWW_{1ij(s)} \leq w_{1i} - w_{1i}^L(1-biW_{1ij(s)}), \ \ ZWW_{1ij(s)} \geq w_{1i} - w_{1i}^U(1-biW_{1ij(s)}) \quad \forall s \in \{1,...,\bar{W}_{1ij}\}
\end{array}\right\} (24)$$

$$\left.\begin{array}{l} WY_{21ij} = \sum_{s=1}^{\bar{W}_{2ij}} W_{2ij(s)} ZWY_{21ij(s)} \\ y_{1j}^{L} biW_{2ij(s)} \le ZWY_{21ij(s)} \le y_{1j}^{U} biW_{2ij(s)},\ \ ZWY_{21ij(s)} \le y_{1j} - y_{1j}^{L}(1-biW_{2ij(s)}),\ \ ZWY_{21ij(s)} \ge y_{1j} - y_{1j}^{U}(1-biW_{2ij(s)}) \quad \forall s \in \{1,...,\bar{W}_{2ij}\} \\ WY_{22ij} = \sum_{s=1}^{\bar{W}_{2ij}} W_{2ij(s)} ZWY_{22ij(s)} \\ y_{2j}^{L} biW_{2ij(s)} \le ZWY_{22ij(s)} \le y_{2j}^{U} biW_{2ij(s)},\ \ ZWY_{22ij(s)} \le y_{2j} - y_{2j}^{L}(1-biW_{2ij(s)}),\ \ ZWY_{22ij(s)} \ge y_{2j} - y_{2j}^{U}(1-biW_{2ij(s)}) \quad \forall s \in \{1,...,\bar{W}_{2ij}\} \\ WW_{2ij} = \sum_{s=1}^{\bar{W}_{2ij}} W_{2ij(s)} ZWW_{2ij(s)} \\ w_{2i}^{L} biW_{2ij(s)} \le ZWW_{2ij(s)} \le w_{2i}^{U} biW_{2ij(s)},\ \ ZWW_{2ij(s)} \le w_{2i} - w_{2i}^{L}(1-biW_{2ij(s)}),\ \ ZWW_{2ij(s)} \ge w_{2i} - w_{2i}^{U}(1-biW_{2ij(s)}) \quad \forall s \in \{1,...,\bar{W}_{2ij}\} \end{array}\right\} \quad (25)$$

By applying these transformation steps, the model (10) can now be transformed into its final formulation as a single level mixed integer linear programming problem as shown in the model (26). The table in the model (26) is used to identify some additional constraints and conditions for adding these constraints to the model (26).

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

$$\begin{array}{lll} \text{s.t.} & \sum_{j} WY_{11ij} \le h_{1i} + \sum_{l|Ind_{il}=1} TX_{11il} + \sum_{l|Ind_{il}=0} T_{1il}^* x_{1l} & \forall i \in L \\ & \sum_{j} WY_{21ij} = h_{2i} + \sum_{l} TX_{21il} & \forall i \in E \\ & \sum_{j} WY_{12ij} \le h_{1i} + \sum_{l|Ind_{il}=1} T_{1il} x_{\Omega l} + \sum_{l|Ind_{il}=0} T_{1il}^* x_{\Omega l} & \forall i \in L \\ & \sum_{j} WY_{22ij} = h_{2i} + \sum_{l} T_{2il} x_{\Omega l} & \forall i \in E \\ & \sum_{i \in L} WW_{1ij} + \sum_{i \in E} WW_{2ij} \ge q_j & \forall j \\ & \multicolumn{2}{l}{\sum_{i \in L} HW_{1i} + \sum_{i \in E} HW_{2i} + \sum_{i \in L}\left(\sum_{l|Ind_{il}=1} TW_{1il} x_{\Omega l} + \sum_{l|Ind_{il}=0} T_{1il}^* w_{1i} x_{\Omega l} \right) + \sum_{i \in E}\left(\sum_{l} TW_{2il} x_{\Omega l} \right) = \sum_{j} QY_{2j}} \\ & y_{1j} \ge 0, y_{2j} \ge 0 \ \ \forall j, \quad w_{1i} \ge 0 \ \ \forall i \in L, \ \ x_{1l} \in \{0,1\} \ \ \forall l & \end{array}$$

Condition to Add the Constraints	Constraint Reference	Constraint Index Set
$Ind_{il} = 1$	(9), (11), and (12)	For all $i \in L$, For all l
Always	(13), (14), and (15)	For all $i \in E$, For all l
Always	(16) and (18)	For all $i \in L$
Always	(17) and (19)	For all $i \in E$
Always	(20) and (21)	For all j
Always	(22) and (24)	For all $i \in L$, For all j
Always	(23) and (25)	For all $i \in E$, For all j

(26)

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-2 model can now be solved as a mixed integer linear program by solving the model (26). The optimal objective function value of the model (26), Δ^{U*}, 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 not satisfied, add scenario ω_2^* which is the optimal settings of $\xi = (\vec{c}, \vec{q}, \vec{h}_1, \vec{h}_2, \boldsymbol{T_1}, \boldsymbol{T_2}, \boldsymbol{W}_1, \boldsymbol{W_2})$ from the model (26) to the scenario set Ω and return to the first stage algorithm. Otherwise, the algorithm is terminated with an $\varepsilon-$optimal robust solution which is the discrete solute on with the maximum regret of Δ^U from the model (26). In fact, we do not have to solve the model (26) to optimality in each iteration to generate the scenario ω_2^*. We can stop the optimization process for the model (26) as soon as the feasible solution with the objective value larger than the current value of Δ^U has been found. We can use this feasible setting of $\xi = (\vec{c}, \vec{q}, \vec{h}_1, \vec{h}_2, \boldsymbol{T_1}, \boldsymbol{T_2}, \boldsymbol{W}_1, \boldsymbol{W_2})$ to generate the scenario ω_2^* in the current iteration. The following Lemma 4 provides the important result that the algorithm always terminates at an ε-optimal robust solution in a finite number of algorithm steps.

***Lemma* 4:** The three-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 an ε-optimal robust solution to the original problem.

Proof: The result follows form 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. □

The following section demonstrates applications of the proposed three-stage algorithm for solving min-max regret robust optimization problems under full-factorial scenario design of data uncertainty. All results illustrate the promising capability of the proposed algorithm for solving the robust optimization of this type with an extremely large number of possible scenarios.

3. Applications of the Three-Stage Algorithm

In this section, we describe numerical experiments using the proposed algorithm for solving a number of two-stage facility location problems under uncertainty. We consider the supply chain in which suppliers send material to factories that supply warehouses that supply markets as shown in Figure 2 (Chopra and Meindl). Location, capacity allocation, and transportation decisions have to be made in order to minimize the overall supply chain cost. 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 demands. The model requires the following notations, parameters and decision variables:

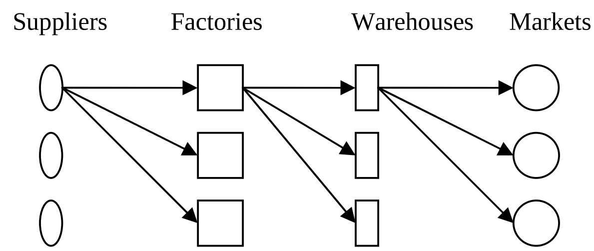

Figure 2: Stages in the Considered Supply Chain Network (Chopra and Meindl).

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
y_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

In the deterministic case, the overall problem can be modeled as the mixed integer linear programming problem presented in the following model. When some parameters in the model are uncertain, the goal becomes to identify robust factory and warehouse locations under deviation robustness definition. Transportation decisions are treated as recourse decisions which will be made after the realization of uncertainty.

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

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

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

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

$$x_{1hi} \ge 0 \;\; \forall h \; \forall i, \; x_{2ie} \ge 0 \;\; \forall i \; \forall e, \; x_{3ej} \ge 0 \;\; \forall e \; \forall j, \; s_j \ge 0 \;\; \forall j, \; y_i \in \{0,1\} \;\; \forall i, \text{ and } z_e \in \{0,1\} \;\; \forall e$$

We apply the proposed algorithm to 20 different experimental settings of the robust facility location problems. Each experimental setting in this case study contains different sets of uncertain parameters and different sets of possible locations which result in different number of possible scenarios. The number of possible scenarios in this case study varies from 64 up to 3^{40} 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. Let us define notations *l'*, *n'*, *t'*, and *m'* to represent the number of suppliers, factories, warehouses, and markets with uncertain parameters respectively in the problem. It is assumed that each uncertain parameter in the model can independently take its values from *r* possible real values. The following Table 1 describes these twenty settings of the case study.

Table 1: Twenty Settings of Numerical Problems in the Case Study

Problem	*l*	*n*	*t*	*m*	*l'*	*n'*	*t'*	*m'*	*r*	$\lvert\bar{\Omega}\rvert$	Size of Extensive Form Model: #Constraints	#Continuous Variables	#Binary Variables
S1	8	8	8	8	0	2	2	2	2	64	3.14×10^{3}	1.13×10^{4}	16
S2	8	8	8	8	0	2	2	4	2	256	1.25×10^{4}	4.51×10^{4}	16
S3	8	8	8	8	0	2	4	2	2	256	1.25×10^{4}	4.51×10^{4}	16
S4	8	8	8	8	0	2	2	6	2	1024	5.02×10^{4}	1.80×10^{5}	16
S5	8	8	8	8	0	2	4	4	2	1024	5.02×10^{4}	1.80×10^{5}	16
S6	8	8	8	8	0	2	6	2	2	1024	5.02×10^{4}	1.80×10^{5}	16
S7	8	8	8	8	0	2	4	6	2	4096	2.01×10^{5}	7.21×10^{5}	16
S8	8	8	8	8	0	2	6	4	2	4096	2.01×10^{5}	7.21×10^{5}	16
S9	8	8	8	8	0	2	6	6	2	2^{14}	8.03×10^{5}	2.88×10^{6}	16
M1	8	8	8	8	0	6	6	6	2	2^{18}	1.28×10^{7}	4.61×10^{7}	16
M2	6	6	6	6	6	6	6	6	3	3^{24}	1.04×10^{13}	2.71×10^{13}	12
M3	6	6	6	6	6	6	6	6	3	3^{24}	1.04×10^{13}	2.71×10^{13}	12
M4	6	6	6	6	6	6	6	6	3	3^{24}	1.04×10^{13}	2.71×10^{13}	12
M5	6	6	6	6	6	6	6	6	3	3^{24}	1.04×10^{13}	2.71×10^{13}	12
M6	6	6	6	6	6	6	6	6	3	3^{24}	1.04×10^{13}	2.71×10^{13}	12
L1	8	8	8	8	8	8	8	8	3	3^{32}	9.08×10^{16}	3.26×10^{17}	16
L2	8	8	8	8	8	8	8	8	3	3^{32}	9.08×10^{16}	3.26×10^{17}	16
L3	8	8	8	8	8	8	8	8	3	3^{32}	9.08×10^{16}	3.26×10^{17}	16
L4	8	8	8	8	8	8	8	8	3	3^{32}	9.08×10^{16}	3.26×10^{17}	16
L5	10	10	10	10	10	10	10	10	3	3^{40}	7.42×10^{20}	3.40×10^{21}	20

All case study settings are solved by the proposed algorithm with ε = 0 and extensive form model (EFM) both without and with Benders' decomposition (BEFM) on a Windows XP-based Pentium(R) 4 CPU 3.60GHz personal computer with 2.00 GB RAM using a C++ program and CPLEX 10 for the optimization process. MS-Access is used for the case study input and output database. In this case study, we apply the proposed algorithm to these twenty experimental problems by using two different setups of initial scenarios. For the first setup, the initial scenario set consists of only one scenario. For the second setup, the initial scenario set consists of all combinations of upper and lower bounds for each main type of uncertain parameters. For example, there are three main types of uncertain parameters in the problems S1 to S9, the initial scenario set of these problems consists of 2^3 = 8 scenarios for the second setup. Table 2 illustrates the computation time (in seconds) and performance comparison among these methodologies over all 20 settings. If the algorithm fails to obtain an optimal robust solution within 24 hours or fails to solve the problem due to insufficiency of memory, the computation time of "--" is reported in the table. Because the problems considered in this case study are always feasible for any setting of location decisions, the Stage2 of the proposed algorithm can be omitted.

Table 2: Performance Comparison between Proposed Algorithm and Traditional Methods

Problem	EFM Solution Time	BEFM Solution Time	Proposed Algorithm							
			Initial Scenarios Setup 1				Initial Scenarios Setup 2			
			Stage1 Time	Stage3 Time	Total Time	#Iteration ($\|\Omega\|$)	Stage1 Time	Stage3 Time	Total Time	#Iteration ($\|\Omega\|$)
S1	80	56.6	9.4	2.7	12.1	4 (5)	5.6	0.9	6.5	1 (9)
S2	1495	178.9	9.9	2.7	12.6	4 (5)	5.8	1.1	6.9	1 (9)
S3	1923	186.4	9.4	2.4	11.8	3 (4)	5.7	1	6.7	1 (9)
S4	27540	743.2	10.6	2.7	13.3	4 (5)	5.8	1.3	7.1	1 (9)
S5	38630	802.4	9.6	1.8	11.4	3 (4)	5.9	0.9	6.8	1 (9)
S6	39757	805.2	9.5	2.8	12.3	4 (5)	11.8	2.2	14	3 (11)
S7	--	5382.8	9.9	1.6	11.5	3 (4)	6.2	1.1	7.3	1 (9)
S8	--	6289.5	10.8	2.7	13.5	4 (5)	12	2.1	14.1	3 (11)
S9	--	64521.6	11.4	2.9	14.3	4 (5)	13.1	2	15.1	3 (11)
M1	--	--	57.2	7.1	64.3	9 (10)	45.2	5.4	50.6	5 (13)
M2	--	--	113.5	33.7	147.2	28 (29)	40	15.7	55.7	10 (26)
M3	--	--	65.9	1469.8	1535.7	17 (18)	39.2	1310.5	1349.7	11 (27)
M4	--	--	57.1	24.5	81.6	15 (16)	21.9	9	30.9	5 (21)
M5	--	--	91.3	1702.4	1793.7	22 (23)	29.3	1201.1	1230.4	8 (24)
M6	--	--	137.5	142.1	279.6	14 (15)	55.6	178.7	234.3	6 (20)
L1	--	--	550	26553	27103	42 (43)	149.9	24394.8	24544.7	17 (33)
L2	--	--	221.2	948	1169.2	19 (20)	69.8	1069.6	1139.4	10 (26)
L3	--	--	548.5	77.1	625.6	35 (36)	158.4	36.2	194.6	11 (27)
L4	--	--	952	12816	13768	49 (50)	110.9	8125.8	8236.7	15 (31)
L5	--	--	51338	33765	85103	87 (88)	17559	29038	46597	75 (91)

All results from these experimental runs illustrate significant improvements in computation time of the proposed algorithm over the extensive form model both with and without Benders' decomposition. These results demonstrate the promising capability of the proposed algorithm for solving practical min-max regret robust optimization problems under full factorial scenario design of data uncertainty with extremely large number of possible scenarios. In addition, these numerical results illustrate the impact of the quality of the initial scenarios setup on the required computation time of the proposed algorithm. Decision makers are highly recommended to perform thorough analysis of the problem in order to construct the good initial set of scenarios before applying the proposed algorithm.

4. Summary

This paper develops a min-max regret robust optimization algorithm for dealing with uncertainty in model parameter values of mixed integer linear programming problems when each uncertain model parameter independently takes its value from a finite set of real numbers with unknown joint probability distribution. This type of parametric uncertainty is referred to as a full-factorial scenario design. The algorithm consists of three stages and coordinates three mathematical programming formulations to solve the overall optimization problem efficiently. The algorithm utilizes pre-processing steps, decomposition techniques, and problem transformation procedures to improve its computational tractability. The algorithm is proven to either terminate at an optimal robust solution, or identify the nonexistence of the robust solution, in a finite number of iterations. The proposed algorithm has been applied to solve a number of robust facility location problems under uncertainty. All results illustrate the outstanding performance of the proposed algorithm for solving the robust optimization problems of this type over the traditional methodologies.

5. References

Assavapokee, T., Realff, M., Ammons, J., and Hong, I.: A Scenario Relaxation Algorithm for Finite Scenario Based Min-Max Regret and Min-Max Relative Regret Robust Optimization. *Computers & Operations Research*, (To appear) (2007a).

Assavapokee, T., Realff, M., and Ammons, J.: A Min-Max Regret Robust Optimization Approach for Interval Data Uncertainty. *Journal of Optimization Theory and Applications*, (To appear) (2007b).

Averbakh, I.: On the Complexity of a Class of Combinatorial Optimization Problems with Uncertainty, *Mathematical Programming*, 90, 263-272 (2001).

Averbakh, I.: Minmax Regret Solutions for Minimax Optimization Problems with Uncertainty, *Operations Research Letters*, 27/2, 57-65 (2000).

Benders, J.F.: Partitioning Procedures for Solving Mixed Variables Programming Problems, *Numerische Mathematik*, v4, 238-252 (1962).

Ben-Tal, A. and Nemirovski, A.: Robust Convex Optimization, *Mathematical Methods of Operations Research*, 23, 769-805 (1998).

Ben-Tal, A. and Nemirovski, A.: Robust Solutions to Uncertain Programs, *Operations Research Letters*, 25, 1-13 (1999).

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

Ben-Tal, A., El-Ghaoui, L., and Nemirovski, A.: Robust Semidefinite Programming, In: Saigal, R., Vandenberghe, L., Wolkowicz, H., (eds.), *Semidefinite Programming and Applications*, Kluwer Academic Publishers (2000).

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

Bertsimas, D. and Sim, M.: The Price of Robustness, *Operations Research*, 52/1, 35-53 (2004).

Chopra, S. and Meindl, P.: Supply Chain Management: Strategy, Planning, and Operations (2nd edition), *Prentice Hall Publishers* (2003).

Kouvelis, P. and Yu, G.: *Robust Discrete Optimization and Its Applications,* Kluwer Academic Publishers, Dordecht, The Netherlands (1997).

Mausser, H.E. and Laguna, M.: A New Mixed Integer Formulation for the Maximum Regret Problem, *International Transactions in Operational Research*, 5/5, 389-403 (1998).

Mausser, H.E. and Laguna, M.: Minimizing the Maximum Relative Regret for Linear Programmes with Interval Objective Function Coefficients, *Journal of the Operational Research Society*, 50/10, 1063-1070 (1999a).

Mausser, H.E. and Laguna, M.: A Heuristic to Minimax Absolute Regret for Linear Programs with Interval Objective Function Coefficients, *Europian Journal of Operational Research*, 117, 157-174 (1999b).

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

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