J Optim Theory Appl (2008) 137: 297–316
DOI 10.1007/s10957-007-9334-6

Min-Max Regret Robust Optimization Approach on Interval Data Uncertainty

T. Assavapokee · M.J. Realff · J.C. Ammons

Published online: 15 December 2007

Abstract This paper presents a three-stage optimization algorithm for solving two-stage deviation robust decision making problems under uncertainty. The structure of the first-stage problem is a mixed integer linear program and the structure of the second-stage problem is a linear program. Each uncertain model parameter can independently take its value from a real compact interval with unknown probability distribution. The algorithm coordinates three mathematical programming formulations to iteratively solve the overall problem. This paper provides the application of the algorithm on the robust facility location problem and a counterexample illustrating the insufficiency of the solution obtained by considering only a finite number of scenarios generated by the endpoints of all intervals.

Keywords Robust optimization · Interval data uncertainty · Min-max regret robust optimization · Deviation robust optimization

1 Introduction and Background

This paper addresses the problem of two-stage decision making under uncertainty, where the uncertainty appears in the values of key parameters of a mixed integer

---

Communicated by P.M. Pardalos.

This work was supported by the National Science Foundation through Grant DMI-0200162.

T. Assavapokee (✉)
Department of Industrial Engineering, University of Houston, Houston, TX, USA
e-mail: tiravat.assavapokee@mail.uh.edu

M.J. Realff
Department of Chemical and Biomolecular Engineering, Georgia Institute of Technology, Atlanta, GA, USA

J.C. Ammons
Department of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA

linear programming (MILP) formulation,

$$\max\{\vec{q}^T\vec{x}+\vec{c}^T\vec{y}|W\vec{x}\le\vec{h}+T\vec{y}, V\vec{x}=\vec{g}+S\vec{y}, \vec{x}\ge\vec{0}, \vec{y}\in\{0,1\}^{|\vec{y}|}\}.$$

In the model formulation, let the vector $\vec{y}$ represent the first-stage decisions that have to be made before the realization of uncertainty and let the vector $\vec{x}$ represent the second-stage decisions that can be made after the realization of uncertainty. Let the vectors $\vec{c}$, $\vec{h}$, $\vec{g}$, $\vec{q}$ and the matrices T, S, W, V represent parameters of the decision model. Each uncertain parameter (except the parameters $\vec{q}$, W, V) can independently take its value from a real compact interval with unknown probability distribution. We assume that the model parameters W and V are deterministic and each element of the parameters $\vec{q}$ can independently take its value from a finite set of real numbers.

Because of the incomplete information about the joint probability distribution of the uncertain parameters in the problem, decision makers are not able to search for the first-stage decisions with the best long run average performance. Instead, decision makers are searching for the robust first-stage decisions that perform well across all possible input scenarios without attempting to assign an assumed probability distribution to any ambiguous parameter. In this paper, we propose an optimization algorithm that can efficiently identify the robust first-stage decisions under the deviation robustness definition in Kouvelis and Yu [1] on interval data uncertainty of two-stage MILP problems. The proposed algorithm sequentially solves and updates a relaxation problem until both the feasibility and optimality conditions of the overall problem are satisfied. The feasibility and optimality verification steps involve the use of bilevel programming, which coordinates a Stackelberg game (Von Stackelberg, [2]) between the decision environment and decision makers; this is explained in detail in Sect. 2. In addition, preprocessing procedures and problem transformation steps are presented for improving the computational tractability of the proposed algorithm. The proposed algorithm is proven to terminate at an optimal deviation robust solution (if one exists) in a finite number of iterations.

Deviation 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 robust 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. Because of this definition, deviation robust optimization is often referred as min-max regret robust optimization. The work of Kouvelis and Yu [1] summarizes the state-of-art in deviation robust 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 [3–6] 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. Averbakh [7, 8] 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 the weights of the elements) when each weight can vary within an interval under the deviation robustness definition. Bertsimas and Sim [9, 10] 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 problems 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. [11] presents an algorithm for solving scenario-based min-max regret and relative regret robust optimization problems for two-stage MILP formulations.

In Sect. 2, we present the theoretical methodology of the proposed algorithm. In Sect. 3, we illustrate some example applications of the proposed algorithm on robust optimization problems with interval data uncertainty.

2 Methodology

This section begins by reviewing key concepts of the scenario based min-max regret robust optimization and the concept of the extensive form formulation. 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 algorithm always terminates at the robust optimal solution (if one exists) in a finite number of iterations.

We address the problem where the basic components of the model uncertainty are represented by a set of all possible scenarios of the input parameters, referred as the scenario set $\bar{\Omega}$. The problem contains two types of decision variables. Let the vector $\vec{y}$ denote binary choice first-stage decision variables and let the vector $\vec{x}_\omega$ denote continuous second-stage recourse decision variables and let vectors $\vec{c}_\omega$, $\vec{q}_\omega$, $\vec{h}_\omega$, $\vec{g}_\omega$ and matrices T_ω, S_ω, W_ω, V_ω denote model parameters setting for each scenario $\omega \in \bar{\Omega}$. If the realization of the model parameter is known to be scenario ω a priori, the optimal choice for the decision variables $(\vec{y}, \vec{x}_\omega)$ can be obtained by solving the following model (1):

$$O_\omega^* = \max\ \vec{q}_\omega^T \vec{x}_\omega + \vec{c}_\omega^T \vec{y}, \tag{1a}$$

$$\text{s.t.}\quad W_\omega \vec{x}_\omega - T_\omega \vec{y} \le \vec{h}_\omega, \tag{1b}$$

$$V_\omega \vec{x}_\omega - S_\omega \vec{y} = \vec{g}_\omega, \tag{1c}$$

$$\vec{x}_\omega \ge \vec{0}, \qquad \vec{y} \in \{0,1\}^{|\vec{y}|}. \tag{1d}$$

When the parameter uncertainty (ambiguity) exists, the search for the robust first-stage solution comprises finding decisions $\vec{y}$ such that the function $\max_{\omega\in\bar{\Omega}}(O_\omega^* - Z_\omega^*(\vec{y}))$ is minimized, where for each scenario $\omega \in \bar{\Omega}$,

$$\begin{aligned} Z_\omega^*(\vec{y}) = \max\ & \vec{q}_\omega^T \vec{x}_\omega + \vec{c}_\omega^T \vec{y}, \\ \text{s.t.}\quad & W_\omega \vec{x}_\omega \le \vec{h}_\omega + T_\omega \vec{y}, \\ & V_\omega \vec{x}_\omega = \vec{g}_\omega + S_\omega \vec{y}, \\ & \vec{x}_\omega \ge \vec{0}. \end{aligned}$$

In the case when the scenario set $\bar{\Omega}$ is a finite set, the optimal choice of the first-stage decision variables $\vec{y}$ can be obtained by solving the following model (2):

$$\min \delta, \tag{2a}$$

$$\text{s.t.} \quad \delta \geq O^*_\omega - \vec{q}^{\,T}_\omega \vec{x}_\omega - \vec{c}^{\,T}_\omega \vec{y}, \quad \forall \omega \in \bar{\Omega}, \tag{2b}$$

$$W_\omega \vec{x}_\omega - T_\omega \vec{y} \leq \vec{h}_\omega, \quad \forall \omega \in \bar{\Omega}, \tag{2c}$$

$$V_\omega \vec{x}_\omega - S_\omega \vec{y} = \vec{g}_\omega, \quad \forall \omega \in \bar{\Omega}, \tag{2d}$$

$$\vec{x}_\omega \geq \vec{0}, \quad \forall \omega \in \bar{\Omega}, \tag{2e}$$

$$\vec{y} \in \{0, 1\}^{|\vec{y}|}. \tag{2f}$$

This model (2) is referred as the extensive form model of the problem. If an optimal solution for the model (2) exists, the resulting optimal setting of the decision variables $\vec{y}$ represents the optimal setting of the robust first-stage decisions. Unfortunately, when the scenario set $\bar{\Omega}$ is an infinite set, the problem cannot be solved directly by using the extensive form model.

We propose the algorithm which can overcome this limitation. Let the vector $\xi = (\vec{c}, T, S, \vec{h}, \vec{g})$ denote the uncertain parameters under interval data uncertainty defining the objective function and constraints of the optimization problem. Under the interval data uncertainty, the scenario set $\bar{\Omega}$ is an infinite set which is generated by all possible values of the parameter vectors ξ and $\vec{q}$. As described below, we propose a three-stage algorithm for solving the deviation robust optimization problem under the scenario set $\bar{\Omega}$.

Proposed Three-Stage Algorithm

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) 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. 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 $Y_\Omega = \vec{y}^*$ (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. Solve the Bilevel-1 model specified in Sect. 2.2 by using the Y_Ω information from Step 2. If the optimal objective function value of the Bilevel-1 model is nonnegative (feasible case), proceed to Step 4. Otherwise (infeasible case), set $\Omega \leftarrow \Omega \cup \{\omega^*_1\}$, where ω^*_1 is the infeasible scenario for Y_Ω generated by the Bilevel-1 model in this iteration and return to Step 1.

Step 4. Solve the Bilevel-2 model specified in Sect. 2.3 by using the Y_Ω information from Step 2. Let ω_2^* and Δ^{U*} denote the scenario with maximum regret value for Y_Ω and the optimal objective function value generated by the Bilevel-2 model respectively in this iteration. Set $\Delta^U \leftarrow \min\{\Delta^{U*}, \Delta^U\}$. 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 $\Omega \leftarrow \Omega \cup \{\omega_2^*\}$, and return to Step 1.

We 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-stage and the third-stage of the algorithm respectively. Each of these three algorithm stages is detailed in the following subsections.

2.1 First-Stage Algorithm

The purposes of the first-stage algorithm are (i) to find the candidate robust decision from a considered finite subset of scenarios Ω, (ii) to find the lower bound for the min-max regret value, and (iii) to determine if the algorithm has discovered a global optimal (or an ε-optimal) robust solution for the problem. The first-stage algorithm utilizes two main optimization models: the model (1) and the relaxed model (2). The algorithm calculates O_ω^* for all scenarios in the (small) finite subset Ω of $\bar{\Omega}$ (as needed). In each iteration, the set of scenarios in Ω is enlarged to include one additional scenario generated by either the second-stage or the third-stage of the algorithm. If the model (1) is infeasible for any generated scenario, the algorithm is terminated with the conclusion that there exists no robust solution to the problem. Otherwise, the optimal objective function values of the model (1) for all scenarios in the set Ω are used as the required parameters O_ω^* in the relaxed model (2).

Once all required values of O_ω^* are obtained $\forall\omega \in \Omega$, the relaxed model (2) is solved by considering only the scenario set Ω. 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 Y_Ω and the lower bound of the min-max regret value Δ^L obtained from the relaxed model (2). The optimality condition, $\Delta^U - \Delta^L \le \varepsilon$, is then checked. If the optimality condition is satisfied, the robust solution associated with Δ^U is the globally ε-optimal robust solution and the algorithm is terminated. Otherwise, the candidate robust solution is forwarded to the second-stage.

2.2 Second-Stage Algorithm

The main purpose of the second stage is to find a parametric scenario in $\bar{\Omega}$ that make the candidate robust decision Y_Ω infeasible in the model (1). To achieve this goal, the algorithm solves a bilevel programming problem referred to as the Bilevel-1 model by following the two main steps. In the first step, some model parameters values in the original Bilevel-1 model are predetermined at their optimal setting by following some simple preprocessing rules. In the second step, the Bilevel-1 model is transformed from its original form into a single-level MILP structure.

One can find a model parameters setting that make the Y_Ω solution infeasible for the model (1) by solving the following bilevel programming problem (Bilevel-1 model). In the Bilevel-1 model, the leader tries to make the problem infeasible (any slack variables negative) by controlling the parameters settings $(T, S, \vec{h}, \vec{g})$. The follower problem tries to make the problem feasible (all slack variables nonnegative) by controlling the continuous decision variables values $(\vec{x}, \vec{s}, \vec{s}_1, \vec{s}_2, \delta)$ under the fixed parameters setting of the leader, when the setting of the binary decision variables is fixed at Y_Ω. The following model (3) demonstrates the general structure of the Bilevel-1 model. Let L and E represent sets of row indices associating with less-than-or-equal-to and equality constraints in the model (1) respectively. In the model (3), δ represents a scalar decision variable and the vectors $\vec{0}$ and $\vec{1}$ represent the vectors with the value of 0 and 1 respectively for all elements in the vectors. If the resulting optimal setting of δ (the minimum value of slack variables) is nonnegative, the candidate solution Y_Ω is guaranteed to be feasible over all possible scenarios in $\bar{\Omega}$. Otherwise, the optimal parameters setting resulting from the Bilevel-1 model represents an infeasible scenario for the model (1) under the fixed first-stage decision vector Y_Ω

$$\min\ \delta, \tag{3a}$$

$$\text{s.t.}\quad h_i^L \le h_i \le h_i^U, \quad \forall i \in L, \tag{3b}$$

$$T_{il}^L \le T_{il} \le T_{il}^U, \quad \forall i \in L, \forall l, \tag{3c}$$

$$g_i^L \le g_i \le g_i^U, \quad \forall i \in E, \tag{3d}$$

$$S_{il}^L \le S_{il} \le S_{il}^U, \quad \forall i \in E, \forall l, \tag{3e}$$

$$\max\ \delta, \tag{3f}$$

$$\text{s.t.}\quad W\vec{x} + \vec{s} = \vec{h} + TY_\Omega, \tag{3g}$$

$$V\vec{x} + \vec{s}_1 = \vec{g} + SY_\Omega, \tag{3h}$$

$$-V\vec{x} + \vec{s}_2 = -\vec{g} - SY_\Omega, \tag{3i}$$

$$\delta\vec{1} \le \vec{s}, \qquad \delta\vec{1} \le \vec{s}_1, \qquad \delta\vec{1} \le \vec{s}_2, \tag{3j}$$

$$\vec{x} \ge \vec{0}. \tag{3k}$$

The current form of the model (3) has a bilevel linear structure. Because the structure of the follower problem of the model (3) is a linear program and it affects the leader decisions only through it objective function, we can simply replace the follower problem of the model (3) with explicit representations of its optimality conditions including its primal, dual, and strong duality constraints. In addition, from the special structure of the model (3), all elements in the decision variables vector $\vec{h}$ and matrix T can be predetermined to either one of their bounds even before solving the model (3).

For each element of the decision vector $\vec{h}$ and matrix T, its optimal setting is the lower bound of its possible values. The correctness of these simple rules is obvious. After applying these simple preprocessing rules, the follower problem transformation, and the result of the following Lemma 2.1, the model (3) can be transformed from a bilevel linear structure to a single-level MILP structure presented in the model (4). These results greatly simplify the solution methodology of the Bilevel-1 model.

Lemma 2.1 *The model* (3) *has at least one optimal solution* S^* *and* $\vec{g}^*$ *in which each element of* S^* *and* $\vec{g}^*$ *takes on a value at one of its bounds.*

Proof Each of these variables S_{il} and g_i appears in only two constrains in the model (3):

$$\sum_j V_{ij}x_j + s_{1i} = g_i + \sum_l S_{il}Y_{\Omega l} \quad \text{and} \quad -\sum_j V_{ij}x_j + s_{2i} = -g_i - \sum_l S_{il}Y_{\Omega l}.$$

It is also easy to see that

$$s_{1i} = -s_{2i} \quad \text{and} \quad \min\{s_{1i}, s_{2i}\} = -|s_{1i} - s_{2i}|/2.$$

This fact implies that the optimal setting of $\vec{x}$ 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

$$|s_{1i} - s_{2i}|/2 = \left| g_i + \sum_l S_{il}Y_{\Omega l} - \sum_j V_{ij}x_j \right|,$$

the optimal setting of S_{il} and g_i will maximize

$$\min_{\vec{x}\in\chi(Y_\Omega)} \left| g_i + \sum_l S_{il}Y_{\Omega l} - \sum_j V_{ij}x_j \right|, \quad \text{where}$$

$$\chi(Y_\Omega) = \{\vec{x} \geq \vec{0} | W\vec{x} \leq \vec{h} + TY_\Omega, V\vec{x} = \vec{g} + SY_\Omega\}.$$

In this form, it is easy to see that the optimal setting of the variables S_{il} and g_i will take on one of their bounds. □

In the model (4), M represents a significantly large real number and w_{1i}, $\forall i \in L$, w_{2i}^+ and w_{2i}^-, $\forall i \in E$ represent dual variables of the follower problem. The decision variables GW_i^+, GW_i^-, SW_{il}^+, SW_{il}^- are used to replace the nonlinear terms $g_i w_{2i}^+$, $g_i w_{2i}^-$, $S_{il}w_{2i}^+$, $S_{il}w_{2i}^-$ in the model respectively. After applying these transformations, the Bilevel-1 model can finally be solved using traditional MILP methods by solving the model (4). The obtained solution is used to decide whether to add scenario ω_1^*, which is the combination of the optimal setting of T, S, $\vec{h}$, $\vec{g}$ from the model (4) and any feasible combination of $\vec{c}$, $\vec{q}$, W, V to the scenario set Ω and return to the first stage (if $\delta^* < 0$), or to forward the values of Y_Ω and Δ^L to the third stage (if $\delta^* \geq 0$)

$$\min\ \delta, \tag{4a}$$

$$\text{s.t.}\quad \sum_j W_{ij}x_j + s_i = h_i^L + \sum_l T_{il}^L Y_{\Omega l},\quad \forall i \in L, \tag{4b}$$

$$\sum_j V_{ij}x_j + s_{1i} = g_i + \sum_l S_{il}Y_{\Omega l},\quad \forall i \in E, \tag{4c}$$

$$-\sum_j V_{ij}x_j + s_{2i} = -g_i - \sum_l S_{il}Y_{\Omega l},\quad \forall i \in E, \tag{4d}$$

$$\left.\begin{array}{l} S_{il} = S_{il}^L + (S_{il}^U - S_{il}^L)(biS_{il}),\\ S_{il}^L w_{2i}^+ \le SW_{il}^+ \le S_{il}^U w_{2i}^+,\\ S_{il}^L w_{2i}^- \le SW_{il}^- \le S_{il}^U w_{2i}^-,\\ S_{il}^U w_{2i}^+ - M(1 - biS_{il}) \le SW_{il}^+ \le S_{il}^L w_{2i}^+ + M(biS_{il}),\\ S_{il}^U w_{2i}^- - M(1 - biS_{il}) \le SW_{il}^- \le S_{il}^L w_{2i}^- + M(biS_{il}),\\ \quad biS_{il} \in \{0,1\}, \end{array}\right\}\ \forall i \in E,\ \forall l, \tag{4e}$$

$$\left.\begin{array}{l} g_i = g_i^L + (g_i^U - g_i^L)(big_i),\\ g_i^L w_{2i}^+ \le GW_i^+ \le g_i^U w_{2i}^+,\\ g_i^L w_{2i}^- \le GW_i^- \le g_i^U w_{2i}^-,\\ g_i^U w_{2i}^+ - M(1 - big_i) \le GW_i^+ \le g_i^L w_{2i}^+ + M(big_i),\\ g_i^U w_{2i}^- - M(1 - big_i) \le GW_i^- \le g_i^L w_{2i}^- + M(big_i),\\ \quad big_i \in \{0,1\}, \end{array}\right\}\ \forall i \in E, \tag{4f}$$

$$\delta \le s_i,\quad \forall i \in L,\qquad \delta \le s_{1i},\quad \forall i \in E,\qquad \delta \le s_{2i},\quad \forall i \in E, \tag{4g}$$

$$\sum_{i\in L} W_{ij}w_{1i} + \sum_{i\in L}(V_{ij}(w_{2i}^+ - w_{2i}^-)) \ge 0,\quad \forall j, \tag{4h}$$

$$\sum_{i\in L} w_{1i} + \sum_{i\in E}(w_{2i}^+ + w_{2i}^-) = 1, \tag{4i}$$

$$\delta = \sum_{i\in L}\left(h_i^L + \sum_l T_{il}^L Y_{\Omega l}\right)w_{1i} + \sum_{i\in E}\left(GW_i^+ - GW_i^- + \sum_l (SW_{il}^+ - SW_{il}^-)Y_{\Omega l}\right), \tag{4j}$$

$$w_{1i} \ge 0,\quad \forall i \in L,\qquad w_{2i}^+ \ge 0,\quad \forall i \in E, \tag{4k}$$

$$w_{2i}^- \ge 0,\quad \forall i \in E,\qquad x_j \ge 0,\quad \forall j. \tag{4l}$$

2.3 Third-Stage Algorithm

The main purpose of the third-stage algorithm is to identify a scenario ω_2^* with the largest regret value for the candidate robust first-stage decision Y_Ω overall scenarios in $\bar{\Omega}$. We look for the scenario ω_2^* such that

$$\omega_2^* \in \arg\max_{\omega\in\bar{\Omega}}\{O_\omega^* - Z_\omega^*(Y_\Omega)\}.$$

The mathematical model utilized by this stage is also a bilevel program referred to as the Bilevel-2 model. The leader problem is tasked with finding the setting of the vector $\bar{\xi} = (\vec{c}, T, S, \vec{h}, \vec{g}, \vec{q})$ and the vector $(\vec{x}_1, \vec{y}_1)$ that result in the maximum regret value overall possible scenarios, $\max_{\omega\in\bar{\Omega}}\{O_\omega^* - Z_\omega^*(Y_\Omega)\}$, for the candidate robust solution Y_Ω. The follower problem on another hand is tasked to respond with the setting of vector $\vec{x}_2$ that maximizes the value of $Z_\omega^*(Y_\Omega)$, under the setting of the vector $\bar{\xi}$ established by the leader problem. The structure of the Bilevel-2 model is represented in the following model (5):

$$\max\ q^T\vec{x}_1 + \vec{c}^T\vec{y}_1 - \vec{q}^T\vec{x}_2 - \vec{c}^T Y_\Omega, \tag{5a}$$

$$\text{s.t.}\quad W\vec{x}_1 \le \vec{h} + T\vec{y}_1, \tag{5b}$$

$$V\vec{x}_1 = \vec{g} + S\vec{y}_1, \tag{5c}$$

$$c_l^L \le c_l \le c_l^U, \quad \forall l, \tag{5d}$$

$$T_{il}^L \le T_{il} \le T_{il}^U, \quad \forall i \in L, \forall l, \tag{5e}$$

$$S_{il}^L \le S_{il} \le S_{il}^U, \quad \forall i \in E, \forall l, \tag{5f}$$

$$h_i^L \le h_i \le h_i^U, \quad \forall i \in L, \tag{5g}$$

$$g_i^L \le g_i \le g_i^U, \quad \forall i \in E, \tag{5h}$$

$$q_j \in \{q_{j(1)}, q_{j(2)}, \dots, q_{j(mj)}\}, \quad \forall j, \tag{5i}$$

$$\vec{x}_1 \ge \vec{0}, \qquad \vec{y}_1 \in \{0,1\}^{|\vec{y}_1|}, \tag{5j}$$

$$\max\ \vec{q}^T\vec{x}_2, \tag{5k}$$

$$\text{s.t.}\quad W\vec{x}_2 \le \vec{h} + TY_\Omega, \tag{5l}$$

$$V\vec{x}_2 = \vec{g} + SY_\Omega, \tag{5m}$$

$$\vec{x}_2 \ge \vec{0}. \tag{5n}$$

The solution methodology for solving the model (5) has the same two steps as solving the Bilevel-1 formulation.

2.3.1 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 $\vec{c}$ is represented in the objective function of the model (5) as $c_l y_{1l} - c_l Y_{\Omega l}$. From any given value of $Y_{\Omega l}$, the value of c_l can be predetermined by the following simple rules. If the value of $Y_{\Omega l}$ 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_{il} of the parameter matrix T is presented in the functional constraints of the model (5) as

$$\sum_j W_{ij}x_{1j} \le h_i + T_{il}y_{1l} + \sum_{k\neq l} T_{ik}y_{1k} \quad \text{and}$$

$$\sum_j W_{ij}x_{2j} \le h_i + T_{il}Y_{\Omega l} + \sum_{k\neq l} T_{ik}Y_{\Omega k}.$$

For any given Y_Ω information, the value of T_{il} can be predetermined at T_{il}^U if the value of $Y_{\Omega l}$ is zero. In the case when the value of $Y_{\Omega l}$ is one, the optimal setting of T_{il} satisfies the following set of constraints illustrated in (6) where the new variable TY_{il} replaces the nonlinear term $T_{il}y_{1l}$ in the model (5). The insight of this set of constraints (6) is that, if the value of y_{1l} is set to be zero by the model, the optimal setting of T_{il} is T_{il}^L. Otherwise, the optimal setting of T_{il} can take any value from the compact interval $[T_{il}^L, T_{il}^U]$

$$TY_{il} - T_{il} + T_{il}^L(1 - y_{1l}) \le 0, \tag{6a}$$

$$-TY_{il} + T_{il} - T_{il}^U(1 - y_{1l}) \le 0, \tag{6b}$$

$$T_{il}^L y_{1l} \le TY_{il} \le T_{il}^U y_{1l}, \tag{6c}$$

$$T_{il} \le T_{il}^L + y_{1l}(T_{il}^U - T_{il}^L), \tag{6d}$$

$$T_{il}^L \le T_{il} \le T_{il}^U. \tag{6e}$$

Preprocessing Step for q: Each element q_j of the parameter vector $\vec{q}$ is presented in the objective function of the model (5) as $q_jx_{1j} - q_jx_{2j}$. Each parameter q_j can independently take its values from the ascending ordered set of real values $\{q_{j(1)}, q_{j(2)}, \dots, q_{j(mj)}\}$, 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 Y_Ω information, in the case where the value of x_{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 QX_{1j} and QX_{2j} to replace the terms q_jx_{1j} and q_jx_{2j} respectively in model (5) and a set of variables and constraints illustrated in (7) to replace the constraint $q_j \in \{q_{j(1)}, q_{j(2)}, \dots, q_{j(mj)}\}$ in the model (5) where x_{rj}^U and x_{rj}^L represent the upper bound and the lower bound of variable x_{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 nonzero

$$q_j = \sum_{s=1}^{mj} q_{j(s)} bi_{j(s)},$$

$$\sum_{s=1}^{mj} bi_{j(s)} = 1,\ bi_{j(s)} \ge 0,\ \forall s \in \{1, 2, \dots, mj\} \text{ and } \bigcup_{\forall s}\{bi_{j(s)}\} \text{ is SOS1}, \tag{7a}$$

$$QX_{1j} = \sum_{s=1}^{mj} q_{j(s)} z_{1j(s)}, \tag{7b}$$

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

$$z_{1j(s)} \ge x_{1j} - x_{1j}^U (1 - bi_{j(s)}), \quad \forall s \in \{1, \dots, mj\}, \tag{7c}$$

$$QX_{2j} = \sum_{s=1}^{mj} q_{j(s)} z_{2j(s)}, \tag{7d}$$

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

$$z_{2j(s)} \ge x_{2j} - x_{2j}^U (1 - bi_{j(s)}), \quad \forall s \in \{1, \dots, mj\}. \tag{7e}$$

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. Two counterexamples are given below. The optimal solution for the first counterexample is $h^* = 5$ with a leader optimal objective function value of five. The leader objective function value is always zero when p_3 is set at either one of its bounds (0 or 10). The optimal solution for the second counterexample is $q^* = 8$ with a leader optimal objective value of 3008. The leader objective values are 3006.5 and 3006 when q is set at its lower and upper bounds respectively.

Counterexample 1

$$\begin{aligned}
\max\ & 2x_{11} + x_{12} + x_{13} - 2x_{21} - x_{22} - x_{23}, \\
\text{s.t.}\ & x_{11} + x_{12} \le 10y_1, \qquad x_{11} \le 5y_2, \\
& x_{12} \le h, \qquad x_{11} \le x_{12}, \\
& x_{11} + x_{13} \le 5y_3, \qquad x_{11}, x_{12}, x_{13} \ge 0, \\
& 0 \le h \le 10, \qquad y_1, y_2, y_3 \in \{0, 1\}, \\
\max\ & 2x_{21} + x_{22} + x_{23}, \\
\text{s.t.}\ & x_{21} + x_{22} \le 10Y_{\Omega 1}, \qquad x_{21} \le 5Y_{\Omega 2}, \\
& x_{22} \le h, \qquad x_{21} \le x_{22}, \qquad x_{21} + x_{23} \le 5Y_{\Omega 3}, \\
& x_{21}, x_{22}, x_{23} \ge 0, \quad \text{where } Y_{\Omega 1} = 1,\ Y_{\Omega 2} = 0,\ Y_{\Omega 3} = 1.
\end{aligned}$$

Counterexample 2

$$\begin{aligned}
\max\ & qx_{11} + 8x_{12} - 1000y_1 - 1000y_2 - 1000y_3 - p_4x_{21} - 8x_{22} + 1000Y_{1\Omega} \\
& + 1000Y_{2\Omega} + 1000Y_{3\Omega}, \\
\text{s.t.}\ & x_{11} \le 6 - y_1, \qquad x_{12} \le 6 - y_2, \\
& x_{11} + x_{12} \le 9 - y_3, \qquad (1/3)x_{11} + x_{12} \le 6, \qquad x_{11} + (1/3)x_{12} \le 6,
\end{aligned}$$

$$
\begin{aligned}
& x_{11}, x_{12}, x_{13} \geq 0, \\
& q \in \{7, 8, 12\}, \qquad y_1, y_2, y_3 \in \{0, 1\}, \\
\max\ & q x_{21} + 8 x_{22}, \\
\text{s.t.}\ & x_{21} \leq 6 - Y_{\Omega 1}, \qquad x_{22} \leq 6 - Y_{\Omega 2}, \\
& x_{21} + x_{22} \leq 9 - Y_{\Omega 3}, \qquad (1/3)x_{21} + x_{22} \leq 6, \qquad x_{21} + (1/3)x_{22} \leq 6, \\
& x_{21}, x_{22}, x_{23} \geq 0, \quad \text{where } Y_{\Omega 1} = 1,\ Y_{\Omega 2} = 1,\ Y_{\Omega 3} = 1.
\end{aligned}
$$

2.3.2 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 the model (8):

$$\max\{\vec{q}^T \vec{x}_1 + \vec{c}^T \vec{y}_1 - \vec{q}^T \vec{x}_2 - \vec{c}^T Y_\Omega\}, \tag{8a}$$

$$\text{s.t.}\quad W\vec{x}_1 - T\vec{y}_1 \leq \vec{h}, \tag{8b}$$

$$V\vec{x}_1 - S\vec{y}_1 = \vec{g}, \tag{8c}$$

$$W\vec{x}_2 - TY_\Omega + \vec{s}_1 = \vec{h}, \tag{8d}$$

$$V\vec{x}_2 - SY_\Omega = \vec{g}, \tag{8e}$$

$$W^T \vec{w}_1 + V^T \vec{w}_2 - \vec{a} = \vec{q}, \tag{8f}$$

$$w_{1i} s_{1i} = 0, \quad \forall i \in L, \tag{8g}$$

$$a_j x_{2j} = 0, \quad \forall j, \tag{8h}$$

$$c_l^L \leq c_l \leq c_l^U, \quad \forall l, \tag{8i}$$

$$T_{il}^L \leq T_{il} \leq T_{il}^U, \quad \forall i \in L,\ \forall l, \tag{8j}$$

$$S_{il}^L \leq S_{il} \leq S_{il}^U, \quad \forall i \in E,\ \forall l, \tag{8k}$$

$$h_i^L \leq h_i \leq h_i^U, \quad \forall i \in L, \tag{8l}$$

$$g_i^L \leq g_i \leq g_i^U, \quad \forall i \in E, \tag{8m}$$

$$q_j \in \{q_{j(1)}, q_{j(2)}, \dots, q_{j(mj)}\}, \quad \forall j, \tag{8n}$$

$$
\begin{aligned}
& \vec{x}_1 \geq \vec{0}, \qquad \vec{x}_2 \geq \vec{0}, \qquad \vec{s}_1 \geq \vec{0}, \\
& \vec{a} \geq \vec{0}, \qquad \vec{w}_1 \geq \vec{0}, \qquad \vec{y}_1 \in \{0, 1\}^{|\vec{y}_1|}.
\end{aligned} \tag{8o}
$$

Finally, the model (8) is transformed into a single level MILP problem with complementary slackness constraints as shown in the model (9) 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 [12] 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 [13] for bilevel programming problems because high numerical precision is required to avoid the leader problem perturbing the follower problem optimal solution. More theoretical concepts on Bilevel and Multilevel programming can be found in Bard and Falk [14], Bard [15, 16], Hansen et al. [17], Migdalas et al. [18], and Heng and Pardalos [19]

$$\max\ \Delta^{U*} = \sum_{j|Ind\_q_j=1} QX_{1j} + \sum_{j|Ind\_q_j=0} q_j^* x_{1j} + \sum_l c_l^* y_{1l} - \sum_{j|Ind\_q_j=1} QX_{2j} - \sum_{j|Ind\_q_j=0} q_j^* x_{2j} - \sum_l c_l^* Y_{\Omega l}, \tag{9a}$$

$$\text{s.t.}\quad \sum_j W_{ij}x_{1j} - \sum_{l|Ind\_T_{il}=1} TY_{il} - \sum_{l|Ind\_T_{il}=0} T_{il}^* y_{1l} \le h_i, \quad \forall i \in L, \tag{9b}$$

$$\sum_j V_{ij}x_{1j} - \sum_l SY_{il} = g_i, \quad \forall i \in E, \tag{9c}$$

$$\sum_j W_{ij}x_{2j} - \sum_{l|Ind\_T_{il}=1} T_{il}Y_{\Omega l} - \sum_{l|Ind\_T_{il}=0} T_{il}^* Y_{\Omega l} + s_{1i} = h_i, \quad \forall i \in L, \tag{9d}$$

$$\sum_j V_{ij}x_{2j} - \sum_l S_{il}Y_{\Omega l} = g_i, \quad \forall i \in E, \tag{9e}$$

$$\sum_{i\in L} W_{ij}w_{1i} + \sum_{i\in E} V_{ij}w_{2i} - a_j = q_j, \quad \forall j \text{ such that } Ind\_q_j = 1, \tag{9f}$$

$$\sum_{i\in L} W_{ij}w_{1i} + \sum_{i\in E} V_{ij}w_{2i} - a_j = q_j^*, \quad \forall j \text{ such that } Ind\_q_j = 0, \tag{9g}$$

$$S_{il}^L y_{1l} \le SY_{il} \le S_{il}^U y_{1l}, \quad \forall i \in E, \forall l, \tag{9h}$$

$$SY_{il} \le S_{il} - S_{il}^L(1 - y_{1l}), \quad \forall i \in E, \forall l, \tag{9i}$$

$$SY_{il} \ge S_{il} - S_{il}^U(1 - y_{1l}), \quad \forall i \in E, \forall l, \tag{9j}$$

$$w_{1i}s_{1i} = 0, \quad \forall i \in L, \tag{9k}$$

$$a_j x_{2j} = 0, \quad \forall j, \tag{9l}$$

$$T_{il}^L \le T_{il} \le T_{il}^U, \quad \forall i \in L,\ \forall l, \tag{9m}$$

$$S_{il}^L \le S_{il} \le S_{il}^U, \quad \forall i \in E,\ \forall l, \tag{9n}$$

$$h_i^L \le h_i \le h_i^U, \quad \forall i \in L, \tag{9o}$$

$$g_i^L \le g_i \le g_i^U, \quad \forall i \in E, \tag{9p}$$

$$x_{1j} \ge 0, \qquad x_{2j} \ge 0, \qquad s_{1i} \ge 0, \qquad a_j \ge 0, \qquad w_{1i} \ge 0,$$
$$y_{1l} \in \{0,1\}, \quad \forall i \in L,\ \forall j,\ \forall l. \tag{9q}$$

Condition to add constraints	Constraint reference	Constraint index set	
$Ind\_T_{il} = 1$	(6a–6d)	For all $i \in L, l$	(9r)
$Ind\_q_j = 1$	(7a–7e)	For all j	(9s)

where

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

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

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

$$q_j^* = \begin{cases} \text{Preprocessed value of } q_j, \ \text{if } q_j \text{ can bepreprocessed,} \\ 0, \quad \text{otherwise.} \end{cases}$$

For any branch and bound scheme, the branching rules are always critical. For the model (9), 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 (9) can be solved. The optimal objective function value Δ^{U*} of the model (9) is used to update the value of Δ^U by setting Δ^U to $\min\{\Delta^{U*}, \Delta^U\}$. The optimality condition is then checked. If the optimality condition is not satisfied, add scenario ω_2^* which is the combination of the optimal settings of $\vec{c}, T, S, \vec{h}, \vec{g}, \vec{q}, W, V$ from the model (9) to the scenario set Ω and return to the first-stage algorithm. Otherwise, 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 (9). The following Lemma 2.2 provides the important result that the algorithm always terminates at a globally ε-optimal robust solution in finite number of algorithm steps.

Lemma 2.2 *The algorithm presented terminates in a finite number of steps. After the algorithm terminates with $\varepsilon = 0$, it has either detected infeasibility or has found an optimal robust solution to the original problem.*

Proof Notice that the relaxed model (2) is a relaxation of the original min-max regret problem and the feasible region of the model (1) contains the feasible region of the original problem. This has four important implications: (a) if the model (1) is infeasible, then the original min-max regret problem is also infeasible, (b) if the relaxed model (2) is infeasible, then the original min-max regret problem is also infeasible, (c) $\Delta^L \le \min_{\vec{y}}(\max_{\omega\in\bar{\Omega}}(O^*_\omega - Z^*_\omega(\vec{y})))$ for all iterations, and (d) if $\vec{y}^*$ is an optimal solution to the original problem, then $\vec{y}^*$is a feasible solution to the relaxed model (2). From the first and second implications, it is clear that if the algorithm terminates because either the model (1) or the relaxed model (2) is infeasible then the original min-max regret problem is infeasible. Now suppose that the algorithm terminates in Step 2 or Step 4 with $\Delta^L = \Delta^U$ and the solution Y_Ω. Notice that we can go to Step 4 only if Y_Ω is a feasible solution to the overall problem, then

$$\Delta^U = \max_{\omega\in\bar{\Omega}}(O^*_\omega - Z^*_\omega(Y_\Omega)) \ge \min_{\vec{y}}\Big(\max_{\omega\in\bar{\Omega}}(O^*_\omega - Z^*_\omega(\vec{y}))\Big).$$

Therefore, if $\Delta^L = \Delta^U$, then

$$\max_{\omega\in\bar{\Omega}}(O^*_\omega - Z^*_\omega(Y_\Omega)) = \min_{\vec{y}}\Big(\max_{\omega\in\bar{\Omega}}(O^*_\omega - Z^*_\omega(\vec{y}))\Big)$$

or Y_Ω is an optimal solution to the original min-max regret problem. Because there are a finite number of possible combinations of Y_Ω and the proposed algorithm generates new value of Y_Ω in each iteration before termination, the proposed three-stage algorithm always terminates in a finite number of steps. It is obvious that the same argument applies for a finite positive ε. □

3 Application and Example Problems of the Proposed Algorithm

In this section, we first present a counterexample illustrating the insufficiency of the robust solution obtained by considering a finite number of scenarios generated by the endpoints of all compact intervals. Finally, we apply the proposed algorithm to a case study on a robust facility location problem for the simplified supply chains network.

3.1 Comparison of Proposed Algorithm and Endpoint Robust Solutions

In this section, we demonstrates a counterexample that illustrates the insufficiency of the robust solution obtained by considering only a finite number of scenarios generated by the endpoints of all intervals under the interval data uncertainty. Let us consider the following decision problem where the parameter h can take any real value between 0 and 10

$$\begin{aligned}
\max\ & 2x_1 + x_2 + x_3 - y_2,\\
\text{s.t.}\ & x_1 + x_2 \le 10y_1,\\
& x_1 \le 5y_2,\\
& x_2 \le h,\\
& x_1 \le x_2,\\
& x_1 + x_3 \le 5y_3,\\
& x_1 \ge 0, \qquad x_2 \ge 0, \qquad x_3 \ge 0,\\
& y_1 \in \{0, 1\}, \qquad y_2 \in \{0, 1\}, \qquad y_3 \in \{0, 1\}.
\end{aligned}$$

The optimal robust solution to the problem, that h can only take its values from its upper or lower bounds, can be obtained by solving the relaxed model (2) with two scenarios ($h = 0$ and $h = 10$). The resulting optimal robust solution by considering only these two scenarios is $(y_1, y_2, y_3) = (1, 0, 1)$ with the min-max regret value of zero. When applying the solution resulting from this approach to the problem that h can take any values from the real compact interval [0, 10], the maximum regret of this solution is actually equal to four when $h = 5$. The proposed algorithm is now applied to this example problem by using two initial scenarios ($h = 0$ and $h = 10$). The first-stage algorithm generates the candidate robust solution by setting $(y_1, y_2, y_3) = (1, 0, 1)$ with the lower bound of zero. This candidate robust solution is then forwarded to the second-stage of the algorithm for the feasibility check. After performing the preprocessing step, the parameter h can be fixed at zero, which has already been considered in the initial scenario set. At the third-stage, the model (9) is solved and the upper bound on the min-max regret value is obtained. Because the upper bound value resulting from this model (9) is 4, which is greater than the lower bound value of zero, the algorithm forwards the setting of h at 5 (scenario 3) and the upper bound of 4 to the first-stage of the algorithm. After one more iteration, the algorithm terminates when the upper and lower bounds are equal at the value of one with the robust solution of $(y_1, y_2, y_3) = (1, 1, 1)$. The maximum regret value of this solution is one. These results illustrate the superiority of the proposed algorithm over the use of a discrete robust optimization algorithm that considers each parameter at its boundaries, which can generate a less than optimal solution and could be misleading for the problem under the interval data uncertainty.

3.2 Application of the Proposed Algorithm to the Robust Facility Location Problem

In this subsection, we apply the algorithm to a hypothetical robust facility location problem under the interval data uncertainty. We consider a supply chain in which suppliers send material to factories that supply warehouses that supply markets [20]. Location decisions have to be made for both the factories and warehouses. Each facility 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 parameters and decision variables:

m number of markets
n number of potential factory locations
l number of suppliers
t number of potential warehouse locations
D_j annual demand from customer j
K_i potential capacity of factory site i
S_h supply capacity at supplier h
W_e potential warehouse capacity at site e
f_{1i} fixed cost of locating a plant at site i
f_{2e} fixed cost of locating a warehouse at site e
c_{1hi} cost of shipping one unit from supplier h to factory i
c_{2ie} cost of shipping one unit from factory i to warehouse e
c_{3ej} cost of shipping one unit from warehouse e to market j

p_j penalty cost per unit of unsatisfied demand at market j
y_i 1, if plant is opened at site i; 0, otherwise
x_{1hi} transportation quantity from supplier h to plant i
x_{3ej} transportation quantity from warehouse e to market j
z_e 1, if warehouse is opened at site e; 0, otherwise
x_{2ie} transportation quantity from plant i to warehouse e
s_j quantity of unsatisfied demand at market j.

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 cost of the system. The overall problem can be formulated as presented in the following model:

$$\begin{aligned}
\min\ & \sum_{i=1}^{n} f_{1i} y_i + \sum_{e=1}^{t} f_{2e} z_e + \sum_{h=1}^{l}\sum_{i=1}^{n} c_{1hi} x_{1hi} \\
& + \sum_{i=1}^{n}\sum_{e=1}^{t} c_{2ie} x_{2ie} + \sum_{e=1}^{t}\sum_{j=1}^{m} c_{3ej} x_{3ej} + \sum_{j=1}^{m} p_j s_j, \\
\text{s.t.}\ & \sum_{i=1}^{n} x_{1hi} \le S_h, \quad \forall h \in \{1,\dots,l\}, \\
& \sum_{h=1}^{l} x_{1hi} - \sum_{e=1}^{t} x_{2ie} = 0, \quad \forall i \in \{1,\dots,n\}, \\
& \sum_{e=1}^{t} x_{2ie} \le K_i y_i, \quad \forall i \in \{1,\dots,n\}, \\
& \sum_{i=1}^{n} x_{2ie} - \sum_{j=1}^{m} x_{3ej} = 0, \quad \forall e \in \{1,\dots,t\}, \\
& \sum_{j=1}^{m} x_{3ej} \le W_e z_e, \quad \forall e \in \{1,\dots,t\}, \\
& \sum_{e=1}^{t} x_{3ej} + s_j = D_j, \quad \forall j \in \{1,\dots,m\}, \\
& x_{1hi} \ge 0, \quad \forall h, \forall i, \qquad x_{2ie} \ge 0, \quad \forall i, \forall e, \qquad x_{3ej} \ge 0, \quad \forall e, \forall j, \\
& s_j \ge 0, \quad \forall j, \qquad y_i \in \{0,1\}, \quad \forall i, \qquad z_e \in \{0,1\}, \quad \forall e.
\end{aligned}$$

When some parameters in this model are uncertain, the goal becomes to identify robust factory and warehouse locations as the first-stage decisions under the deviation robustness definition. Transportation decisions are now the recourse second-stage decisions, which can be made after all model parameters values are realized. Table 1

Table 1 Approximated parameters information

Supplier (h)	S_h	Factory (i)	f_{1i}	K_i	Warehouse (e)	f_{2e}	W_e	Market (j)	D_j	p_j
San Diego	2500	Seattle	75000	2800	Sacramento	37500	2500	Portland	800	125
Denver	3000	San Franc	75000	2400	Oklahoma City	37500	2600	LA	1050	140
Kansas City	2000	Salt Lake	75000	2500	Lincoln	37500	2500	Phoenix	600	120
El Paso	1000	Wilmington	60000	2150	Nashville	40500	3000	Houston	1800	150
Cincinnati	800	Dallas	75000	2700	Cleveland	37500	2600	Miami	1500	125
Boise	500	Minneapolis	60000	2100	Fort Worth	36000	2100	New York	1250	130
Austin	1100	Detroit	81000	3000	Eugene	37500	2300	St. Louis	1050	120

summarizes information on the approximated parameters values associated with suppliers, factories, warehouses, and markets. A variable transportation cost of \$0.01 per unit per mile is assumed in this example.

The key uncertain parameters that we consider in this example are the supply quantity at each supplier, the potential capacity at each factory and warehouse, and finally the penalty cost per unit of unsatisfied demand at each market. We assume that each uncertain parameter, except the penalty cost, can take its values from 80% up to 120% of its approximated value reported in Table 1. Each uncertain unit penalty cost can take its values from 80%, 100%, or 120% of its approximated value. In summary, this example contains 21 uncertain parameters each of which can take its value form a real compact interval and 7 uncertain parameters each of which can take its value from a finite set of three real values.

The case study is solved by utilizing the proposed algorithm with $\varepsilon = 0$ and the initial scenario set Ω containing 16 scenarios on a Windows XP-based Pentium(R) 4 CPU 3.60 GHz personal computer with 2.00 GB RAM using a C++ program and CPLEX 10 for the optimization process. The algorithm terminates at an optimal robust solution within 6 iterations. The algorithm recommends opening production facilities at San Francisco, Dallas, Minneapolis, and Detroit and recommends opening warehouse facilities at Sacramento, Nashville, Cleveland, and Fort Worth with the maximum regret value of \$106,675.54. The total computation time required is 48 minutes and 59 seconds. Figure 1 illustrates the convergence of the upper and lower bounds on the min-max regret produced by the algorithm.

Finally, we compare the performance of the robust solution with the optimal solution from the deterministic model when each parameter is fixed at the value reported in Table 1. This deterministic solution recommends opening production facilities at San Francisco, Dallas, and Detroit and recommends opening warehouse facilities at Sacramento, Oklahoma city, and Nashville with the maximum regret value of \$212,455.02 (\$105,779.48 or 99.16% increase in the maximum regret value from the robust solution). These results illustrate the significant impact of uncertainty on the performance of the long-term decisions. They illustrate the applicability and effectiveness of the proposed algorithm.

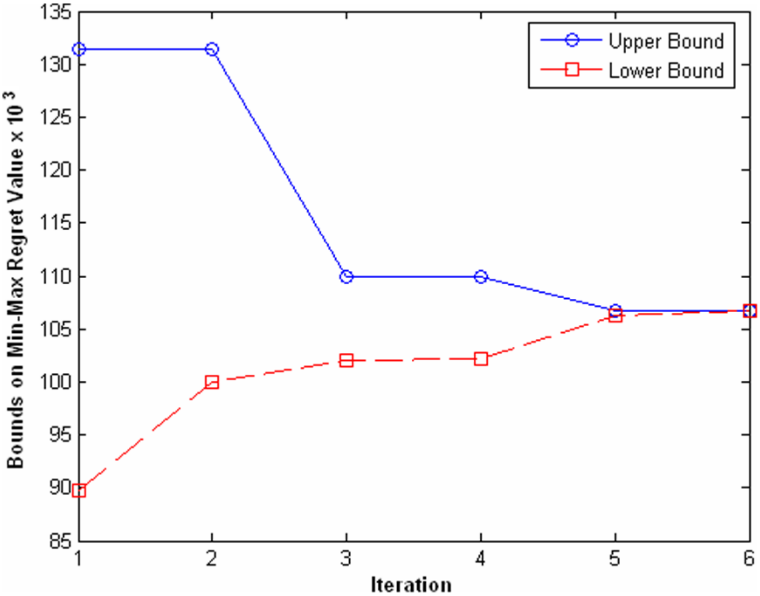

Fig. 1 Upper and lower bounds on min-max regret value from the algorithm

4 Summary

This paper develops a deviation robust optimization algorithm for dealing with uncertainty in model parameter values of MILP problems. The presented algorithm efficiently generates the deviation robust solution to the problem when each uncertain parameter in a general MILP problem takes its value from a real compact interval. The algorithm utilizes preprocessing steps and problem transformation procedures to improve its computational performance. 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. At the end, the paper presents a case study and example illustrating the applicability of the algorithm to practical optimization problems.

References

1. Kouvelis, P., Yu, G.: Robust Discrete Optimization and Its Applications. Kluwer Academic, Dordrecht (1997)
2. Von Stackelberg, H.: Grundzuge der Theoretischen Volkswirtschaftslehre Stuttgart. Kohlhammer, Berlin (1943)
3. Ben-Tal, A., Nemirovski, A.: Robust convex optimization. Math. Methods Oper. Res. **23**, 769–805 (1998)
4. Ben-Tal, A., Nemirovski, A.: Robust solutions to uncertain programs. Oper. Res. Lett. **25**, 1–13 (1999)
5. Ben-Tal, A., Nemirovski, A.: Robust solutions of linear programming problems contaminated with uncertain data. Math. Program. **88**, 411–424 (2000)
6. Ben-Tal, A., El-Ghaoui, L., Nemirovski, A.: Robust semidefinite programming. In: Saigal, R., Vandenberghe, L., Wolkowicz, H. (eds.) Semidefinite Programming and Applications. Kluwer Academic, Dordrecht (2000)

7. Averbakh, I.: Minmax regret solutions for minimax optimization problems with uncertainty. Oper. Res. Lett. **27**(2), 57–65 (2000)
8. Averbakh, I.: On the complexity of a class of combinatorial optimization problems with uncertainty. Math. Program. **90**, 263–272 (2001)
9. Bertsimas, D., Sim, M.: Robust discrete optimization and network flows. Math. Program. Ser. B **98**, 49–71 (2003)
10. Bertsimas, D., Sim, M.: The price of robustness. Oper. Res. **52**(1), 35–53 (2004)
11. Assavapokee, T., Realff, M., Ammons, J., Hong, I.: Scenario relaxation algorithm for finite scenario based min-max regret and min-max relative regret robust optimization. Comput. Oper. Res. **35**(6), (2008)
12. Bard, J.F., Moore, J.T.: A branch and bound algorithm for the bilevel programming problem. SIAM J. Sci. Stat. Comput. **11**(2), 281–292 (1990)
13. Assavapokee, T.: Semi continuous robust approach for strategic infrastructure planning of reverse production systems. Ph.D. Thesis, Georgia Institute of Technology (2004)
14. Bard, J.F., Falk, J.E.: An explicit solution to the multi-level programming problem. Comput. Oper. Res. **9**(1), 77–100 (1982)
15. Bard, J.F.: Some properties of bilevel programming problem. J. Optim. Theory Appl. **68**(2), 371–378 (1991)
16. Bard, J.F.: Practical Bilevel Optimization: Algorithms and Applications. Nonconvex Optimization and Its Applications. Kluwer Academic, Dordrecht (1998)
17. Hansen, P., Jaumard, B., Savard, G.: New branch-and-bound rules for linear bilevel programming. SIAM J. Sci. Stat. Comput. **13**(5), 1194–1217 (1992)
18. Migdalas, A., Pardalos, P.M., Varbrand, P. (eds.): Multilevel Optimization: Algorithms and Applications. Kluwer Academic, Dordrecht (1997)
19. Huang, H.X., Pardalos, P.M.: A multivariate partition approach to optimization problems. Cybern. Syst. Anal. **38**(2), 265–275 (2002)
20. Chopra, S., Meindl, P.: Supply Chain Management: Strategy, Planning, and Operations, 2nd edn. Prentice Hall, Upper Saddle River (2003)