ROBUST SENSITIVITY ANALYSIS FOR LINEAR PROGRAMMING WITH ELLIPSOIDAL PERTURBATION

RUOTIAN GAO AND WENXUN XING*

Department of Mathematical Sciences
Tsinghua University
Beijing 100086, China

(Communicated by Jie Sun)

ABSTRACT. Sensitivity analysis is applied to the robust linear programming problem in this paper. The coefficients of the linear program are assumed to be perturbed in three perturbation manners within ellipsoidal sets. Our robust sensitivity analysis is to calculate the maximal radii of the perturbation sets to keep some properties of the robust feasible set. Mathematical models are formulated for the robust sensitivity analysis problems and all models are either reformulated into linear programs or convex quadratic programs except for the bi-convex programs where more than one row of the constraint matrix is perturbed. For the bi-convex programs, we develop a binary search algorithm.

1. Introduction. Generally, the robust counterpart of linear programming is formulated as follows:

\[
\begin{align*}
\text{min} & \quad c^T x \\
\text{s.t.} & \quad Ax \leq b, \ \forall (A, b) \in U(l) \\
& \quad x \geq 0,
\end{align*}
\]

where \( x \in \mathbb{R}^n \) is the decision variable, \( c \in \mathbb{R}^n \) denotes the objective vector, \( (A, b) \in \mathbb{R}^{m \times n} \times \mathbb{R}^m \) denotes the coefficients perturbed in a given ellipsoidal uncertainty set \( U(l) \) and \( l \) denotes the ellipsoidal radius or radii of the uncertainty set depending on different perturbation manners considered later. The reason why we emphasize \( l \) in \( U(l) \) is that we intend to let this fixed parameter \( l \) in concept of the robust optimization become a variable in our sensitivity analysis. Denote the feasible set of (1) as

\[
\mathcal{F}(l) = \{ x | Ax \leq b, \ \forall (A, b) \in U(l), x \geq 0 \}.
\]

Without loss of generality, we do not consider the uncertainty of the objective function in (1) since we can equivalently reformulate the problem by introducing a new variable \( t \), replacing the objective with \( \min t \) and adding a new constraint \( c^T x \leq t \). And we do not consider the simultaneous perturbation of the matrix \( A \)}
and the right-hand-side vector $b$ either, because it is equivalent to the following reformulation:

$$\min \quad c^T x$$

$$s.t. \quad Ax + by \leq 0, \quad \forall (A, b) \in U(l)$$

$$y = -1$$

$$x \geq 0,$$

where perturbation only occurs in the left part of the constraints.

In this paper, our sensitivity analysis for the robust linear programming problem, which is also called robust sensitivity analysis for linear programming (RSALP), is to get the maximal perturbation radius ($l$) to ensure some properties of the robust feasible set $F(l)$ of (1), where $\bar{l}$ is a given fixed nonnegative parameter, can be kept in $F(l)$ when $l \geq \bar{l}$. In this paper, we assume that (1) is feasible when the uncertainty set is $U(\bar{l})$. We use a deterministic polytope $S = \{x \in \mathbb{R}^n | Ex \leq f\}$ with given $E \in \mathbb{R}^{w \times n}, f \in \mathbb{R}^m$ to represent the properties of $x$ we expect to be maintained in $F(l)$, for example, some entries of $x$ must be zero. And we hope $S \cap F(l) \neq \emptyset$ which means there exists at least one robust feasible solution in $F(l)$ satisfying such properties. Then the model is formulated as follows:

$$\max \quad l$$

$$s.t. \quad Ax \leq b, \quad \forall (A, b) \in U(l)$$

$$x \in S$$

$$l \geq \bar{l}$$

$$x \geq 0, l \geq 0,$$

where $x, l$ are decision variables and $l \geq \bar{l}$ is defined component-wise. It is assumed that $S \cap F(l) \neq \emptyset$, then the model (2) must be feasible.

We use the activity-analysis problem [9] to interpret the motivation of our model. The nonnegative production level $x \in \mathbb{R}^n_+$ consumes resources $Ax \in \mathbb{R}^m$ respecting the resource supply $b \in \mathbb{R}^m$. For a given current activity level $x^*, x^*_i > 0$ means the $i$-th item is indeed produced. We set $P = \{j | x^*_j = 0\}$ as the set of indices corresponding to current zero entries. In real application, the class of items produced should be maintained in a certain time period because it is difficult to react timely owing to some new setup costs.

For example, let us consider the resource $b$ fluctuating in a time period by the market. Based on the perturbation data in hand, we find the uncertainty set is in an ellipsoidal form, and then (1) is formulated with the uncertain data in an ellipsoidal set for a fixed radius $\bar{l}$. An activity level $x^*$ was scheduled before the time period. Ideally, the manager expects the activity level not to change for the time period, then the property set is $S = \{x^*\}$. Our model gives the maximal perturbation radius $l^*(l^* \geq \bar{l})$ of the perturbation of $b$ which keeps the feasibility of $x^*$. If $l^* > \bar{l}$, then $x^*$ can be kept when $b$ is perturbed in this period in a larger uncertainty set $U(l^*)$. Or the manager can select the property set $S = \{x|x_j = 0, j \in P\}$ in our model expecting the items unproduced before not to be produced in this period.

In this paper, the uncertainty set $U(l)$ is considered in the following three ellipsoidal manners in this paper: $b$ is perturbed holistically, $A$ is perturbed row-wisely and holistically, respectively, leading to the following three different forms.

**Case 1. $b$-perturbation.** For any $l \in \mathbb{R}_+$,

$$U(l) = \{A_0\} \times U_b = \{A_0\} \times \left\{b \in \mathbb{R}^m \mid b = b_0 + \sum_{j=1}^\ell \beta_j v_j, \|\beta\| \leq l\right\}. \quad (3)$$
Case 2. A-row-wise perturbation. For any \( l = (l_1, l_2, \ldots, l_m)^T \in \mathbb{R}_+^m \),
\[
U(l) = U_R \times \{b_0\} = \left\{ A \in \mathbb{R}^{m \times n} \mid A^t = A_0^t + (\sum_{j=1}^{n_1} \alpha_j^t u_j^t)^T, \|\alpha^t\| \leq l_i, \; i = 1, \ldots, m \right\} \times \{b_0\}.
\]

(4)

Case 3. A-holistic perturbation. For any \( l \in \mathbb{R}_+ \),
\[
U(l) = U_H \times \{b_0\} = \left\{ A \in \mathbb{R}^{m \times n} \mid A = A_0 + \sum_{j=1}^{s} \gamma_j A_j, \|\gamma\| \leq l \right\} \times \{b_0\}.
\]

(5)

In the above three cases, \( \beta = (\beta_1, \ldots, \beta_t)^T \in \mathbb{R}^t, \alpha^i = (\alpha_1^i, \ldots, \alpha_n^i)^T \in \mathbb{R}^n, i = 1, \ldots, m \) and \( \gamma = (\gamma_1, \ldots, \gamma_s)^T \in \mathbb{R}^s \) are perturbation parameters. \( v_j \in \mathbb{R}^m, j = 1, \ldots, t \) and \( u_j^t \in \mathbb{R}^n, j = 1, \ldots, n_1; i = 1, \ldots, m \) are given perturbation vectors, and \( A_j \in \mathbb{R}^{m \times n}, j = 1, \ldots, s \) are given perturbation matrices. \( A_0^t \) is the \( i \)-th row of \( A_0 \), \( A^t \) is the \( i \)-th row of \( A \) and \( \| \cdot \| \) refers to 2-norm \( \| \cdot \|_2 \). Causing no confusion, we omit the subscript 2. And all the vectors mentioned in this paper are column vectors. All the perturbation sets considered are in ellipsoidal forms controlled by a nonnegative scalar or a nonnegative \( m \)-dimensional vector \( l \), called the “perturbation radius (radii)”. So we do sensitivity analysis for the uncertain data by its radius (radii). Actually, when the uncertainty set is of polytope form, similar conclusions can be obtained by the methods provided in this paper. The model (2) is a multi-objective optimization problem for the A-row-wise perturbation case while is a single-objective one for the other two cases.

From the view of computability, the set \( S \) is a polytope of variable \( x \) in line with linear programming in this paper. The simplest pattern is that \( S = \{x^*\} \) is a singleton set where \( x^* \in F(l) \) is a pre-decision. In this situation, \( E = (I, -I)^T \) and\( f = ((x^*)^T, -(x^*)^T)^T \) where \( I \in \mathbb{R}^{n \times n} \) is an identity matrix. Then (2) yields the maximum radius (radii) when \( x^* \) is still in \( F(l) \). Particularly, if \( x^* \) is optimal to (1) with \( l = \bar{l} \), then it keeps not only feasible but also optimal, see details in Section 3. And \( S \) can be a general polytope containing more than one point, for example, to keep some \( x \)-variables zero, it is described by \( \{x \in \mathbb{R}^n \mid x_j = 0, j \in \mathcal{P}\} \) where \( \mathcal{P} \) is a given index set.

The concept of robust sensitivity analysis comes from the evaluation for a pre-decision persisting with optimality for an uncertain system. For this pre-decision or some properties of this pre-decision, we want to know the maximal perturbation set in which the pre-decision or its properties remain. For example, in the case of \( b \)-perturbation, \( x^* \) is an optimal solution of (1) with \( l = \bar{l} \geq 0 \), can this solution still be optimal for some \( l^* > \bar{l} \)? Of course, it would be the best if \( x^* \) is still optimal under \( b \)-perturbation with the radius \( l^* \) since we need not change the decision at all.

In the classical sensitivity analysis for linear programming, the ranges of the objective coefficients and the ranges of the right-hand-side vector are analyzed to keep the optimal solution or the optimal basis unchanged, respectively, which can be easily found in books about linear programming. In both situations, the ranges of the coefficients are given in a box form.

Robust optimization which came from the robust control community (refer to [7] and [16]) dates back to the work in [14], and has developed rapidly in the past four decades, see e.g., [6], [2] and the references therein. The construction of uncertainty sets, the performance assessment of robust optimal solutions and the computational complexity of the models are three main topics in robust optimization modeling. With poor uncertainty sets, robust models may be computationally intractable or
their robust solutions are too conservative. So how to choose a good uncertainty set is crucial. Based on the basic concept of robust optimization, the uncertainty set(s) should contain all the observed data. So the easiest one is an interval providing the minimum and the maximum of the data for each individual parameter. And the uncertainty sets like ellipsoids or polytopes are then used to describe those data. For interval sets, Bertsimas and Sim [5] found the traditional robust optimization model is rather conservative because it is sensitive to the uncertain data and even has no feasible solution sometimes. They gave a modified model with a probabilistic guarantee of feasibility for its robust optimal solution by counting partial uncertain data of the interval form. With the same concerns on feasibility, Bertsimas and Brown [3] established convex uncertainty sets like polytopes and ellipsoids by using distortion measures to evaluate the feasibility. All models in [3] and [5] preserve computational tractability which are polynomially solvable. But any robust optimal solution of the models in [3] and [5] may not be feasible with a probability for the original robust problems, which somehow does not satisfy the concept of robust optimization. Motivated by the growing availability of data, Bertsimas et al [4] proposed a scheme to design uncertainty sets by using statistical hypothesis tests. And by utilizing asymptotic confidence levels, Ben-Tal et al [1] discussed the construction of uncertainty sets as confidence sets based on φ-divergences when the uncertain parameters contain elements of a probability vector. Furthermore, for those intractable robust counterparts, approximation methods are considered. Marandi et al [12] derived tractable inner and outer approximations for quadratic constraints with convex uncertainty whose robust counterpart is generally intractable.

Xu and Burer [15] presented copositive programs to state the best-case and worst-case optimal values when the parameters in the objective function and the right-hand side are perturbed. For fixed parameters, they got the optimal solution and its optimal value which are not robustly optimal for the perturbation problem although they called their work as “robust sensitivity analysis of the optimal value of linear programming”. They developed tight, tractable conic relaxations to provide the lower and upper bounds.

Our main contributions are stated as follows. First, to the best of our knowledge, this robust sensitivity analysis concept is first brought up and can be applied to evaluate the robustness of a pre-decision. Second, we establish a unified model and give analysis in three different perturbation manners. Generally the radius (radii) in robust optimization is assumed fixed and then is a decision variable in our consideration. Third, the programs in this paper are either linear or bi-convex. As for the bi-convex programming problem with only one row perturbed, it is equivalently reformulated into a convex quadratic programming problem. However, for those bi-convex programs where more than one row is perturbed, we haven’t found equivalent polynomial-time solvable reformulations and thus provide a binary search algorithm, which further triggers our interest in the bi-convex optimization.

The rest of the paper is organized as follows. In Section 2, we analyze the model (2) in three different perturbation manners. In Section 3, we have a discussion on the selection of the pre-decision. In Section 4, the numerical experiment is showed for our algorithm. In addition, one note and the conclusions are provided in Section 5 and Section 6 respectively.

2. The maximum radii. This section gives equivalent reformulations of (2) to get the maximum radius (radii) for the three different perturbation cases. To solve (2),
the key point is that how to describe the semi-infinite constraint $Ax \leq b$, $\forall (A, b) \in U(l)$ and how to find equivalent tractable models. In this section, $x^*$ is supposed to be a given pre-decision feasible for (1) with $l = \bar{l}$.

2.1. $b$-perturbation. In the $b$-perturbation case (3), for any $l \geq \bar{l}$ with given $\bar{l} \in \mathbb{R}_+$,

$$x \in \mathcal{F}(l) \iff A_0x \leq b_0 + \sum_{j=1}^{t} \beta_j v_j, \; \forall \|\beta\| \leq l, \text{ and } x \geq 0.$$ 

Let $v_i \in \mathbb{R}^l$ consist of all the $i$-th entries of $v_j$, $j = 1, \cdots, t$. Then

$$x \in \mathcal{F}(l) \iff (A_0x - b_0)_i \leq \min_{\|\beta\| \leq l} \beta_j v_i, \; i = 1, \cdots, m, \text{ and } x \geq 0$$

$$\iff (A_0x - b_0)_i \leq -l\|v_i\|, \; i = 1, \cdots, m, \text{ and } x \geq 0.$$ 

Denote $v = (\|v_1\|, \cdots, \|v_m\|)^T \in \mathbb{R}_+^m$. Then $\mathcal{F}(l) = \{x | A_0x \leq b_0 - lv, x \geq 0\}$ and (2) becomes a linear programming problem as follows:

$$\max \; l$$

s.t. $A_0x \leq b_0 - lv$

$$l \geq \bar{l}$$

$$x \in \mathcal{S}, \; l \in \mathbb{R}_+.$$ 

Particularly, when $\mathcal{S} = \{x^*\}$, we can write down the maximum radius of (6) directly in the following theorem.

**Theorem 2.1.** Denote $l^*$ as the maximum radius of (6) and $J = \{i | \|v_i\| > 0\}$. In the case of $\mathcal{S} = \{x^*\}$, (i) $l^* - \bar{l} = \min_{i \in J} (b_0^i - \bar{l}\|v_i\| - A_0^i x^*)/\|v_i\| \geq 0$ if $J \neq \emptyset$ or (ii) $l^* = \infty$ if $J = \emptyset$.

**Remark 1.** From the above theorem, we get $l^* - \bar{l} = 0$ if there exists $i$ such that $(A_0x^* - b_0)_i = -\bar{l}\|v_i\|$ and $\|v_i\| > 0$. In this case, any larger perturbation is not allowed. This is a general case to a robustly optimal solution $x^*$ of the robust linear programming problem (1). How to reconstruct the system (1) to get a more robust $x^*$? We shall discuss this issue in Section 3.

2.2. $A$-row-wise perturbation. By (4), for any $l \geq \bar{l}$ with fixed $\bar{l} \in \mathbb{R}_+^m$,

$$x \in \mathcal{F}(l) \iff (A_0^i + \sum_{j=1}^{n_i} \alpha^i_j u^i_j)^T x \leq b_0^i, \; \forall \|\alpha^i\| \leq l_i, \; i = 1, \cdots, m, \text{ and } x \geq 0$$

where $\alpha^i = (\alpha^i_1, \cdots, \alpha^i_{n_i})^T \in \mathbb{R}^{n_i}$ and $b_0^i$ is the $i$-th entry of $b_0$. Denote $H_i = (u^i_1, \cdots, u^i_{n_i})^T \in \mathbb{R}^{n_i \times n}$. Then

$$x \in \mathcal{F}(l) \iff A_0^i x + \max_{\|\alpha^i\| \leq l_i} \left( \alpha^i \right)^T H_i x \leq b_0^i, \; i = 1, \cdots, m, \text{ and } x \geq 0$$

$$\iff A_0^i x + l_i \|H_i\| x \leq b_0^i, \; i = 1, \cdots, m, \text{ and } x \geq 0.$$
Therefore \( F(l) = \{ x \mid A_i^0 x + l_i \|H_i x\| \leq b_i^0, \; i = 1, \cdots, m, x \geq 0 \} \) where \( l = (l_1, l_2, \cdots, l_m)^T \in \mathbb{R}_+^m \) and (2) becomes:

\[
\begin{align*}
\text{max} \; & l \\
\text{s.t.} \; & A_i^0 x + l_i \|H_i x\| \leq b_i^0, \; i = 1, \cdots, m \\
& x \in S \\
& l \geq \bar{l} \\
& x \in \mathbb{R}_+^n, l \in \mathbb{R}_+^m.
\end{align*}
\]

Obviously, (7) is no longer easy because of the cross terms \( l_i \|H_i x\| \), \( i = 1, \cdots, m \).

If \( S = \{ x^\ast \} \), then (7) is a linearly constrained multi-objective programming problem. Since \( l_i, \; i = 1, \cdots, m \) are independent, maximizing \( l \) is equivalent to maximizing all \( l_i, \; i = 1, \cdots, m \) individually. Then we can divide (7) into \( m \) subproblems and treat the following subproblem for each constraint \( i \),

\[
\begin{align*}
\text{max} \; & l_i \\
\text{s.t.} \; & A_i^0 x^\ast + l_i \|H_i x^\ast\| \leq b_i^0 \\
& l_i \geq \bar{l}_i, l_i \in \mathbb{R}_+.
\end{align*}
\]

**Theorem 2.2.** For (8), the maximum radius \( l_i^\ast = +\infty \) if \( \|H_i x^\ast\| = 0 \), otherwise \( l_i^\ast - \bar{l}_i = (b_i^0 - A_i^0 x^\ast - \bar{l}_i \|H_i x^\ast\|)/\|H_i x^\ast\| \geq 0 \). And \( l^\ast = (l_1^\ast, \cdots, l_m^\ast)^T \) is an absolute optimal solution to the multi-objective programming problem (7) i.e., for any feasible solution \( l \), we have \( l^\ast \geq l \).

Next we consider (7) for a general polytope case \( S = \{ x \in \mathbb{R}_+^n | Ex \leq f \} \) where \( E \in \mathbb{R}_+^{w \times n} \) and \( f \in \mathbb{R}_+^w \) are given. We start with a special situation where only one row is perturbed, which is to be proved polynomially computable. Assuming only the \( k \)-th row is perturbed, (7) is in the following form

\[
\begin{align*}
\text{max} \; & l_k \\
\text{s.t.} \; & A_0 x \leq b_0 \\
& A_k^0 x + l_k \|H_k x\| \leq b_k^0 \\
& E x \leq f \\
& l_k \geq \bar{l}_k \\
& x \in \mathbb{R}_+^n, l_k \in \mathbb{R}_+.
\end{align*}
\]

**Remark 2.** The feasible region of (9) is nonempty if and only if (9) is feasible with \( l_k = \bar{l}_k \).

**Remark 3.** For convenience, we do not consider the constraint \( l_k \geq \bar{l}_k \) in subsequent analysis for (9). If the optimal value \( l_k^\ast \) we get without constraint \( l_k \geq \bar{l}_k \) satisfies \( l_k^\ast \geq \bar{l}_k \), then we know (9) is feasible and the optimal value is the same as the one when we retain the constraint. Otherwise (9) is infeasible.

**Proposition 1.** The optimal value of (9) is infinite if the following linear system is feasible,

\[
\{ H_k x = 0, A_0 x \leq b_0, E x \leq f, x \in \mathbb{R}_+^n \}
\] (10)

**Proof.** If the linear system is feasible, then there exists an \( \bar{x} \) such that \( A_0 \bar{x} \leq b_0, \; E\bar{x} \leq f, \; \bar{x} \in \mathbb{R}_+^n \) and \( H_k \bar{x} = 0 \). Thus \( (\bar{x}, l) \) is feasible for (9) for any \( l \geq 0 \), leading to an infinite optimal value of (9). \( \square \)
Remark 4. If the set \( \{ A_0 x \leq b_0, Ex \leq f, x \in \mathbb{R}^n_+ \} = \mathcal{F}(0) \cap \mathcal{S} \) is bounded, then the converse of Proposition 1 also holds.

Proof. If the optimal value of (9) is infinite, let \( \{ l^{(p)}, p = 1, \ldots \} \) be a positive series with \( l^{(p)} \to +\infty \). Then for any \( l^{(p)} \), there exists \( x^{(p)} \) such that \( E x^{(p)} \leq f \), \( x^{(p)} \geq 0 \), and \( A_i x^{(p)} \leq b_0^i \), \( i \neq k \) and \( \| H_k x^{(p)} \| \leq (b_0^k - A_0^k x^{(p)}) / l^{(p)} \). Since \( \{ x^{(p)}, p = 1, \ldots \} \) is bounded, then there exists \( M > 0 \) such that \( b_0^k - A_0^k x^{(p)} \leq M \) holds for any \( x^{(p)}, p = 1, \ldots \). Now we get \( \| H_k x^{(p)} \| \leq M / l^{(p)} \). Because of the boundedness of \( x^{(p)}, p = 1, \ldots \), we can select a convergent subsequence from \( \{ x^{(p)}, p = 1, \ldots \} \), which is also denoted by \( \{ x^{(p)}, p = 1, \ldots \} \) for convenience with \( \| H_k x^{(p)} \| \to 0 \) and \( x^{(p)} \to x^0 \). It is obvious that \( x^0 \) is feasible for the linear system. \( \square \)

If the set \( \{ A_0 x \leq b_0, Ex \leq f, x \in \mathbb{R}^n_+ \} \) is unbounded, then the converse of Proposition 1 does not hold necessarily. Here we are to give an example.

Example 1. Let \( A_0 = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}, b_0 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, E = \begin{pmatrix} 0 & 0 \end{pmatrix}, f = 0, H_2 = \begin{pmatrix} 0 & -1 \end{pmatrix} \)

Then the set \( \mathcal{F}(0) \cap \mathcal{S} = \{ (x_1, x_2)|x_1 \geq 0, x_2 \geq 1 \} \) is unbounded. We only perturb the second row of \( A_0 \) here. The problem (9) is formulated as

\[
\begin{align*}
\max & \quad l_2 \\
\text{s.t.} & \quad -x_1 + l_2 x_2 \leq 0, \\
& \quad x_1 \geq 0, x_2 \geq 1.
\end{align*}
\]

Then \( (x_1, x_2, l_2) = (l^2 + 1, l, l) \) is always a feasible solution if \( l \geq 1 \). Hence the optimal value is +\( \infty \). However, the linear system \( \{-x_2 = 0, x_1 \geq 0, x_2 \geq 1\} \) is infeasible.

Problem (9) is a bi-convex program [10] as it is a second-order cone program when \( l_k \) is fixed and is a linear program when \( x \) is fixed. Generally a bi-convex optimization problem is intractable. But specially (9) is a fractional programming problem and can be transformed into a convex quadratic program equivalently, see the following theorem. The fractional programming problems were studied in [13] and the idea of the proof below can be found therein.

Theorem 2.3. (1) The optimal value of (9) is infinite (i.e., the perturbation can be arbitrarily large) if the linear system (10) is feasible. (2) The optimal value is 0 (i.e., the problem can not be perturbed at all) if and only if the linear system (10) is infeasible and \( b_0^k - A_0^k x = 0 \) holds for any \( x \in \{ x \in \mathbb{R}^n_+ | A_0 x \leq b_0, Ex \leq f \} \). (3) If the linear system (10) is infeasible and there exists an \( x \in \{ x \in \mathbb{R}^n_+ | A_0 x \leq b_0, Ex \leq f \} \) such that \( b_0^k - A_0^k x > 0 \), then the following convex quadratic programming problem gives the square of the reciprocal of the optimal value of (9).

\[
\begin{align*}
\min & \quad \| H_k z \|^2 \\
\text{s.t.} & \quad -A_0^k z + b_0^k t = 1 \\
& \quad A_0 z \leq b_0 t \\
& \quad Ez \leq ft \\
& \quad z \in \mathbb{R}^n_+, \quad t \in \mathbb{R}_+,
\end{align*}
\]

where \( z, t \) are variables.
Proof. The first two claims are obvious. If the linear system (10) is infeasible, then we have \( \|H_k x\| > 0 \) for any \( x \in \{ A_0 x \leq b_0, Ex \leq f, x \in \mathbb{R}_+^n \} \), thus \( \|H_k x\| > 0 \) holds for any feasible solution \( x \) of (9) as \( \{ A_0 x \leq b_0, Ex \leq f, x \in \mathbb{R}_+^n \} \) concludes the feasible domain of (9). Therefore the following fractional program is well defined and is equivalent to (9).

\[
\begin{align*}
\text{max} & \quad y/\|H_k x\| \\
\text{s.t.} & \quad y = b_0^k - A_0^k x \\
& \quad A_0 x \leq b_0 \\
& \quad Ex \leq f \\
& \quad y \in \mathbb{R}_+, x \in \mathbb{R}_+^n.
\end{align*}
\]

Reverse the fractional objective as below:

\[
\begin{align*}
\text{min} & \quad \|H_k x\|/y \\
\text{s.t.} & \quad y = b_0^k - A_0^k x \\
& \quad A_0 x \leq b_0 \\
& \quad Ex \leq f \\
& \quad y \in \mathbb{R}_+, x \in \mathbb{R}_+^n.
\end{align*}
\]

Here we define that \( \alpha/0 = +\infty \) for \( \alpha > 0 \). Next we are to show the square of the optimal value of (12) is equal to the optimal value of (11). Obviously these two problems are both feasible.

For any feasible solution \( (z, t) \) of (11), if \( t = 0 \), then \( (x, y) = (x_0 + dz, y_0 + d) \) is feasible for (12) for any \( d \geq 0 \). Here \( (x_0, y_0) \) is a feasible solution to (12). And \( \|H_k x\|/y = \| \frac{1}{y_0 + d} H_k x + \frac{d}{y_0 + d} H_k z \| \to \| H_k z \| \) as \( d \to +\infty \). It means we can always find a feasible solution of (12) with objective approaching \( \| H_k z \| \) as closely as possible. If \( t > 0 \), then \( (x, y) = (z/t, 1/t) \) is well defined and is feasible for (12) with \( \|H_k x\|/y = \|H_k z\| \). Then the optimal value of (12) is not larger than the positive square root of the optimal value of (11).

For any feasible solution \( (x, y) \) of (12) with \( y \neq 0 \), \( (z, t) = (x/y, 1/y) \) is well defined and is feasible for (11) with \( \|H_k z\| = \|H_k x\|/y \). When \( y = 0 \), \( \|H_k x\|/y \) is infinite and \( (z, t) = (z_0 + dx, t_0 + d) \) is feasible for (11) for any \( d \geq 0 \) where \( (z_0, t_0) \) is a feasible solution to (11). And we have \( \|H_k z\| = \|H_k (z_0 + dx)\| \geq \|H_k z_0\| - d\|H_k x\| \to +\infty \) as \( d \to +\infty \). It follows that the positive square root of the optimal value of (11) is not larger than the optimal value of (12).

Therefore the square of the optimal value of (12) is equal to the optimal value of (11).

Remark 5. The optimal value of (11) is attainable, see Frank-Wolfe theorem [8].

We have not found an equivalent polynomial-time solvable reformulation when \( S \) is a general polytope and more than one row of \( A \) is perturbed. But we can design an algorithm to solve it. Generally, there does not necessarily exist an absolute optimal solution to a multi-objective programming, refer to [11]. We illustrate that problem (7) may not have an absolute optimal solution using the example below.

Example 2. An instance of (7) is formulated as follows:

\[
\begin{align*}
\text{max} & \quad l \\
\text{s.t.} & \quad 3x_1 + 4x_2 + x_3 + l_1\|H_1 x\| \leq 2
\end{align*}
\]
where
\[ H_1 = \begin{pmatrix} 0.1 & 0.1 & 0.2 \\ 0.3 & 0.2 & -0.1 \end{pmatrix}, \quad H_2 = \begin{pmatrix} -0.1 & -0.2 & 0.1 \end{pmatrix}, \]
\[ l = (l_1, l_2)^T \] and \( \bar{l} = (0, 0)^T \). Here let \( S = \{ x \mid x_3 = 0 \} \). First, we only perturb the first row, namely let \( l_2 = 0 \). Then (13) with \( l_2 = 0 \) can be reformed into a convex quadratic programming problem, which has been showed in Theorem 2.3 and the maximal value of \( l_1 \) is 8.9443. Similarly, the maximal value of \( l_2 \) is 5.0000 when \( l_1 \) is fixed 0. Henceforth, if an absolute optimal solution \( \hat{l} \) exists, then there must be \( \hat{l} \geq (8.9443, 0)^T \) and \( \hat{l} \geq (0, 5.0000)^T \). However, the most promising choice \( \hat{l} = (8.9443, 5.0000)^T \) fails to be feasible for (13).

As (7) is a multi-objective optimization problem, we tried to use the techniques of transferring a multi-objective function into a single-objective one and got an optimization problem with only one variable radius remained, but no polynomially computable result was obtained. Two techniques are tried: one is letting \( l = l_k g \) where \( g = (g_1, g_2, \ldots, g_m)^T \in \mathbb{R}^m_+ \) is a given vector with \( g_k = 1 \) and the objective be max \( l_k \), and the other is letting the objective be max \( \min_{i=1,\cdots,m} l_i \).

For the case of \( l = l_k g \) with given \( g \in \mathbb{R}^m_+ \) and variable \( l_k \in \mathbb{R}_+ \), we consider the following bi-convex programming problem
\[
\begin{align*}
& \text{max } l_k \\
& \text{s.t. } A_k^i x + g_i l_k \|H_i x\| \leq b_0^i, \quad i = 1, \cdots, m \\
& \quad \quad Ex \leq f \\
& \quad \quad x \in \mathbb{R}^n_+, l_k \in \mathbb{R}_+. \tag{14}
\end{align*}
\]

Remark 6. In the above model, we do not add the constraint \( l = l_k g \geq \bar{l} \). With the same arguments of Remark 3, we can check it after computation of (14).

We have the following proposition which is a generalized version of Proposition 1.

**Proposition 2.** The optimal value of (14) is infinite if the following linear system is feasible.
\[
\{ Hx = 0, A_0x \leq b_0, Ex \leq f, x \in \mathbb{R}^n_+ \} \tag{15}
\]
Here \( H \in \mathbb{R}^{(\sum_{i \in \text{Supp} \cap n}) \times n} \) consists of \( H_i, \ i \in \text{Supp} \) in turn where \( \text{Supp} = \{ i \mid g_i > 0 \} \).

Similarly, if the set \( \{ A_0x \leq b_0, Ex \leq f, x \in \mathbb{R}^n_+ \} \) is bounded, then the converse also holds. The proof is almost the same so is omitted here.

**Proposition 3.** The optimal value of (14) is 0 if and only if for any \( x \in \{ A_0x \leq b_0, Ex \leq f, x \in \mathbb{R}^n_+ \} \), there exists an \( i \in \text{Supp} \) such that \( H_i x \neq 0 \) and \( b_0^i - A_0^i x = 0 \) hold.

If the condition above does not hold, then we know the optimal value is strictly larger than 0 and thus we can provide a positive lower bound for the optimal value.

**Remark 7.** If the optimal value of (14) is strictly larger than 0, then there exists \( \bar{x} \in \{ A_0x \leq b_0, Ex \leq f, x \in \mathbb{R}^n_+ \} \) such that either \( H_i \bar{x} = 0 \) or \( b_0^i - A_0^i \bar{x} > 0 \) holds for any \( i \in \text{Supp} \). Denote \( W = \{ i \mid H_i \bar{x} \neq 0 \} \). If \( \text{Supp} \cap W \neq 0 \), then let
\[
l_b = \min \{ (b_0^i - A_0^i \bar{x})/(g_i \| H_i \bar{x} \|), \ i \in \text{Supp} \cap W \}, \tag{16}
\]
and obviously $l_b > 0$ is feasible for (14). If $\text{Supp} \cap W = \emptyset$, then $(\bar{x}, l)$ is feasible for (14) for any $l > 0$. Therefore, $l_b$ provides a positive lower bound.

Obviously, the feasible region of $l_b$ in (14) is connected and 0 is a feasible solution. Notice that once we obtain such a point $\bar{x}$, we can derive a positive lower bound $l_b$ for the optimal value. For the convenience of discussion and computation, we assume that there exists $\bar{x}$ such that $A_0\bar{x} < b_0$, $E\bar{x} \leq f$ and $\bar{x} \in \mathbb{R}^n_+$ to assure the maximum perturbation radius is strictly positive. We are now in a position to design an algorithm as follows:

**Algorithm:**

**Step1:** If the linear system (15) is feasible, then the maximal radius is $l_k = +\infty$ and goto Step4, otherwise goto Step2.

**Step2:** Obtain a point $\bar{x} \in \mathbb{R}^n_+$ with $A_0\bar{x} < b_0$, $E\bar{x} \leq f$ and a positive lower bound $l_b$ for the maximal radius which is given by (16). Let $p = 0$ and $q = 1/l_b$.

**Step3:** Let $l_k = 2/(p + q)$ in (14). If (14) is feasible, then let $q = 1/l_k$, otherwise let $p = 1/l_k$. When $q - p < \varepsilon$, goto Step4, otherwise continue this step.

**Step4:** Output the maximal radius $l_k$.

*To obtain such a point $\bar{x}$, we just need to solve (14) with $l_k = 0$. Since the built-in algorithm in MATLAB for linear programming is interior point algorithm and the objective is constant 0, it will return the analytical center of the feasible region, which is a relative interior point.*

Here $\varepsilon$ is a given precision. The number of iterations is no more than $\lceil \log_2(1/(l_b\varepsilon)) \rceil$ and in each iteration it costs polynomial time to solve a second-order cone programming problem.

Under the second technique, we get

\[
\begin{align*}
\max & \quad l_0 \\
\text{s.t.} & \quad A_i^0 x + l_i \|H_i x\| \leq b_0^i, \quad i = 1, \cdots, m \\
& \quad E x \leq f \\
& \quad l_i \geq l_0, \quad i = 1, \cdots, m \\
& \quad x \in \mathbb{R}^n_+, l_0 \in \mathbb{R}_+.
\end{align*}
\]

(17)

By the Remark 8 stated below, we can design almost the same algorithm to solve (17).

**Remark 8.** When $l_0$ is fixed, (17) is feasible if and only if (17) is feasible with $l_i = l_0$, $i = 1, \cdots, m$.

**Remark 9.** When $g = (1, \cdots, 1)^T$, (14) and (17) have the same optimal value.

2.3. A-holistic perturbation. In this case,

\[
\mathcal{F}(l) = \left\{ x \in \mathbb{R}^n_+ | (A_0 + \sum_{j=1}^s \gamma_j A_j)x \leq b_0, \forall \|\gamma\| \leq l \right\},
\]
Similarly, we can get $F$ where $\gamma$ corresponding model to (2) becomes:

$$\text{Theorem 2.4.} \quad \text{For } A \in \mathbb{R}^{n \times n}, \text{ in the case of } S = \{x^*\}, \text{ the maximal radius } l^* = +\infty \text{ if } ||U_i x^*|| = 0, \forall i = 1, \cdots, m, \text{ otherwise } l^* - l = \min_{i \in K}(b_0 - l)||U_i x^*|| - A_0 x^*/||U_i x^*|| \geq 0, \text{ where } K = \{i | ||U_i x^*|| > 0\}.

It is interesting that the problem (18) looks similar to the problem (7). Actually the problem (18) where $A$ is perturbed holistically is a specialization of (7) where $A$ is perturbed row-wisely. In (7), if we let every row have $s$ perturbation vectors and the perturbation direction set of the $j$-th row consist of all the $j$-th rows of $A_i$, $i = 1, \cdots, s$, and additionally force all the perturbation radii to be equal, then the problem (7) is the same as (18). Henceforth, we omit the analysis for this case here.

At the end of this section, we provide a concrete example to demonstrate our results.

**Example 3.** To simplify the analysis, let the fixed $\bar{l} = 0$. The original problem is as follows:

$$\begin{align*}
\min & \quad -2x_1 - x_2 + 2x_3 \\
\text{s.t.} & \quad 3x_1 + 4x_2 + x_3 \leq 2, \\
& \quad -x_1 - 3x_2 - 2x_3 \leq -1, \\
& \quad x_1, x_2, x_3 \geq 0.
\end{align*}$$

Here we let $S$ be two special forms which have been mentioned in Section 1. The first one is $S = \{x^*\}$ where $x^* = (0.05, 0.4, 0)^T$ is a pre-decision and the second one is $S = \{x | x_3 = 0\}$. Notice that $x^*$ is in the latter polytope. Then the maximal radius in the second form is supposed to be larger.

In the $b$-perturbation case, let $v_1 = (0.2, -0.1)^T$ and $v_2 = (0.1, 0.1)^T$ be two perturbation vectors for the right-hand-side vector. Then solving (6) yields the maximal radius $L_1 = 1.1180$ for $S = \{x^*\}$. It means $x^* = (0.05, 0.4, 0)^T$ keeps feasible as long as $l \leq 1.1180$ in (1), alternatively, $x^*$ is infeasible. And for $S = \{x | x_3 = 0\}$, we get the maximal radius $L_2 = 1.6175$. It follows that $L_2 > L_1$.

In the $A$-row-wise perturbation case, let $\alpha_1 = (0.1, 0.1, 0.2)^T$ and $\alpha_2 = (0.3, 0.2, -0.1)^T$ be two perturbation vectors for the first row and $\alpha_3 = (-0.1, -0.2, 0.1)^T$ be the perturbation vector for the second row respectively. Then for $S = \{x^*\}$, the maximal radii $L_3 = (2.3783, 2.9412)^T$. When $l \leq L_3$ in (1), $x^* = (0.05, 0.4, 0)^T$ is feasible and the maximal radii can not be enlarged. For $S = \{x | x_3 = 0\}$, using the first technique, we reformulate problem (13) into a single-objective one and let $g = (1, 1)^T$ to make the radii of the two rows equal. With the help of the binary search algorithm, we can get the maximal radius $L_4 = 3.2106$ for both two rows.
And if we let the two radii equal in the first form, then the maximal radius is 2.3783, which is less than $L_4$.

In the $A$-holistic perturbation case, let two perturbation matrices be

$$A_1 = \begin{pmatrix} 0.2 & -0.1 & 0.1 \\ 0.1 & 0.2 & -0.1 \end{pmatrix} \quad \text{and} \quad A_2 = \begin{pmatrix} -0.1 & 0.3 & -0.2 \\ -0.1 & 0.1 & 0.2 \end{pmatrix},$$

then the maximal radius $L_5 = 1.2657$ for $S = \{x^*\}$. Similarly, we can get the maximal radius $L_6 = 2.6229$ for $S = \{x \mid x_3 = 0\}$. As what we have anticipated, $L_6 > L_5$ holds.

3. Discussions on the pre-decision. Section 2 deals with (2) and yields the maximal perturbation radius (radii) when the polytope $S$ is given. In this section, we give a discussion on how the pre-decision set $S$ affects the maximal perturbation radius and show the tradeoff between the optimality of the pre-decision and the robustness.

3.1. Optimality preference. Generally, we first solve (1) and get an optimal solution $x^*$ with which the value $c^T x^*$ is the minimum for any $l \geq \bar{l}$ by the following proposition.

**Proposition 4.** $\mathcal{F}(l^2) \subseteq \mathcal{F}(l^1)$ whenever $l^2 \geq l^1$. If $x^*$ is a robustly optimal solution of (1), $x^*$ keeps robustly optimal for $l \geq \bar{l}$ if and only if $x^* \in \mathcal{F}(l)$.

**Proof.** The first claim is obvious. Since $\mathcal{F}(l)$ shrinks as $l$ increases, for any $x \in \mathcal{F}(l) \subset \mathcal{F}(\bar{l})$ with $l \geq \bar{l}$, $c^T x \geq c^T x^*$ must hold. If $x^* \in \mathcal{F}(l)$, then $x^*$ is robustly optimal for $l$. Thus the second claim is true.

For sensitivity analysis, one way is to keep this optimal solution unchanged, which gets the optimal value $c^T x^*$ of (1). By setting $S = \{x^*\}$ or $S = \{x \mid c^T x \geq c^T x^*\}$, we do keep the optimality of value $c^T x^*$ of (1).

To simplify our discussions, we only consider the case of setting $\bar{l} = 0$. Similar results can be got for the case $\bar{l} \geq 0$. According to the linear programming theory, generally there exists at least one active row constraint of $A_0 x^* \leq b_0$, i.e., there exists at least one $i$ such that $A_{0i} x^* = b_{0i}$. Then no more robustness $l > \bar{l} = 0$ exists such that $x^* \in \mathcal{F}(l)$ with $l > \bar{l} = 0$ due to Theorem 2.1, Theorem 2.2 and Theorem 2.4. But this may not be absolute. One extreme situation is that all rows in $A x^* \leq b$ are inactive or the active ones are not involved in perturbation, only in which the maximal perturbation radius is positive. An example is shown as below.

**Example 4.**

$$\min \quad x_1 + x_2$$
$$\text{s.t.} \quad x_1 \leq 1, \quad x_2 \leq 1, \quad x_1, x_2 \geq 0.$$  

Obviously, the optimal solution is $x^* = (0, 0)^T$ and the corresponding maximal radius is positive no matter how the problem is perturbed.

When a larger perturbation is not available, one way is to relax the right-hand sides $b_{0i}$ to $b_{0i} + \delta_i$ for all the active constraints in $A_0 x^* \leq b_0$ where the tolerance $\delta_i > 0$, then we get a larger perturbation $l > \bar{l} = 0$ to keep the optimal value $c^T x^*$.  

The other way is to relax the singleton point set \( \{x^*\} \) to a more flexible set \( \mathcal{S} \) which may have more robustness for data perturbation with \( l > l^* \). Different aims are followed by different relaxations. To keep the optimal value not far from \( c^T x^* \), we can relax \( \mathcal{S} = \{x^*\} \) to \( \mathcal{S} = \{x \in \mathbb{R}^n | c^T x \leq c^T x^* + \delta \} \) where \( \delta > 0 \). To keep less new setup costs, we can relax it to \( \mathcal{S} = \{x \in \mathbb{R}^n | x_j = 0, j \in \mathcal{P} \} \) where \( \mathcal{P} \subseteq \mathcal{I} = \{j| x_j^* = 0\} \). These relaxation sets are polytopes and their corresponding robust sensitivity analysis models have been studied in Section 2.

3.2. Robustness preference. Relaxing \( \mathcal{S} \) to \( \mathbb{R}^n \) thoroughly, we solve (2) to get an optimal solution \((x^*, l^*)\). Then \( x^* \) is the most robust with the greatest maximal perturbation radius (radii) \( l^* \) if we let \( x^* \) be a pre-decision.

Different selections of the pre-decision set \( \mathcal{S} \) may affect the maximal perturbation radius (radii) \( l^* \). In our model (2), the pre-decision set is deterministic.

4. Numerical experiments. We are to implement the algorithm for problem (14) with \( l = 0 \) in this section. Whether all entries of \( g \) are equal or not has no effect on the computational complexity of the problem, so we take \( g = (1, \ldots, 1)^T \) for simplicity. It is realized by MATLAB_R2015b on a laptop with Intel Core i5, CPU 2.7GHz and 8G memory. CVX-64 (version 2.0) (http://cvxr.com/cvx/) is used to solve all convex programming problems here.

Since the concrete form of the general polytope will not affect the complexity, we take \( \mathcal{S} = \{x | x_j = 0, j \in \mathcal{P} \} \) here and the cardinality of the index set \( \mathcal{P} \) is given as 30\% of \( n \). And \( \bar{x} \) is supposed to be a strict interior point with \( \bar{x}_j = 0, j \in \mathcal{P} \) and \( \bar{x}_j = 1, j \notin \mathcal{P} \). \( A_0 \) and \( c \) are uniformly distributed in \([-10, 10]^{m \times n}\) and \([-10, 10]^n\) respectively. And \( b_0 - A_0 \bar{x} \) is uniformly distributed in \([2, 10]^n\) to ensure \( A_0 \bar{x} < b_0 \) holds strictly. Here symbol \( |A_0| \) refers to conditional distribution. Besides, we impose a box constraint \( x \in [-100, 100]^n \) to make the feasible region bounded. Then the feasibility of linear system (15) is a sufficient and necessary condition for an infinite perturbation radius. The number \( n_i \) of perturbation vectors for the \( i \)-th row is set in two levels, 25\% and 50\% of \( n + m \). And perturbation vectors \( u^i_j, j = 1, \ldots, n; i = 1, \ldots, m \) are independently uniformly distributed in \([-0.5, 0.5]^n\) or \([-1, 1]^n\). \((m, n)\) is set to be \((20, 20), (20, 100)\) and \((80, 100)\), and the precision \( \epsilon \) is 10\(^{-6}\).

We will show the improvement from the lower bound (16) to the maximum radius, and the effect on CPU time of different combination of \((m, n, pert, scale)\), where \( m \) is the number of constraints, \( n \) is the dimension of the variable, \( pert \) is the perturbation ratio — 25\% or 50\%, and \( scale \) is the perturbation scale of \( u^i_j \) — 0.5 or 1. Here we let the numbers of perturbation vectors of all the rows be equal for simplicity. In every setting \((m, n, pert, scale)\), we do 10 random instances repeatedly. \( Ave \) and \( Std \) stand for the average value and the standard deviation of the 10 experiments.

For these numerical experiments in Table 1, infinite perturbation radius doesn’t occur since it is very rare and special. We can see that the CPU time increases as the scale of the problem \((m, n)\) increases, and \( m \) plays a more important part than \( n \). Besides, the more the perturbation vectors are, the more CPU time it costs whereas the scale of perturbation vectors basically has no effect on CPU time. As for the maximal radius, it is mainly affected by the scale of perturbation vectors. It is easily found that the maximal radius of level \([-0.5, 0.5]^n\) is approximately twice as large as that of level \([-1, 1]^n\). And the other three factors matter little. The
The table below presents numerical results for different instances, including lower bounds, maximum radius improvements, and CPU times. The table columns are labeled as follows:

- **Instance** (m, n, pert, scale)
- **Lower Bound** (Ave, Std)
- **Maximum Radius** (Ave, Std)
- **Improvement Ratio** (Ave, Std)
- **CPU Time** (Ave, Std)

| Instance (m, n, pert, scale) | Lower Bound Ave | Lower Bound Std | Maximum Radius Ave | Maximum Radius Std | Improvement Ratio Ave | Improvement Ratio Std | CPU Time Ave | CPU Time Std |
|-----------------------------|-----------------|-----------------|--------------------|--------------------|-----------------------|-----------------------|--------------|--------------|
| (20, 20, 0.25, 0.5)         | 0.4151          | 0.1551          | 1.1650             | 0.8647             | 2.3001                | 2.4415                | 15.8480      | 1.1491       |
| (20, 20, 0.25, 1)           | 0.2283          | 0.0969          | 0.7672             | 0.3066             | 2.0969                | 1.3452                | 15.6160      | 0.9467       |
| (20, 20, 0.5, 0.5)          | 0.3056          | 0.1443          | 1.0726             | 0.7539             | 2.8557                | 2.0440                | 17.0120      | 1.2904       |
| (20, 20, 0.5, 1)            | 0.1385          | 0.0818          | 0.4476             | 0.2327             | 2.6981                | 1.3452                | 15.5160      | 0.9467       |
| (20, 100, 0.25, 0.5)        | 0.1108          | 0.0106          | 0.5120             | 0.1750             | 3.5828                | 1.5280                | 27.2940      | 1.0984       |
| (20, 100, 0.25, 1)          | 0.0521          | 0.0047          | 0.3152             | 0.2321             | 5.0011                | 4.4017                | 27.3905      | 1.7346       |
| (20, 100, 0.5, 0.5)         | 0.0800          | 0.0086          | 0.4237             | 0.2145             | 2.6969                | 1.3452                | 27.9500      | 1.7346       |
| (20, 100, 0.5, 1)           | 0.0435          | 0.0064          | 0.1897             | 0.2614             | 2.6969                | 1.3452                | 28.1790      | 1.8375       |
| (80, 100, 0.25, 0.5)        | 0.1381          | 0.0534          | 0.5341             | 0.2676             | 2.9914                | 1.9738                | 118.4410     | 5.8674       |
| (80, 100, 0.25, 1)          | 0.0676          | 0.0148          | 0.4115             | 0.1058             | 5.3278                | 1.8796                | 120.0080     | 7.0440       |
| (80, 100, 0.5, 0.5)         | 0.1033          | 0.0238          | 0.5436             | 0.2278             | 4.3561                | 2.7105                | 33.4490      | 1.9935       |
| (80, 100, 0.5, 1)           | 0.0450          | 0.0065          | 0.2548             | 0.0952             | 4.7366                | 2.3818                | 201.1790     | 9.2721       |

1. number of constraints
2. dimension of x
3. ratio of the number of perturbation vectors for one row to m + n
4. scale of perturbation vectors
5. improvement ratio=(maximum radius-lower bound)/lower bound

Table 1. Numerical Results

The improvement ratio is about between 2 and 6, which shows that a naive lower bound is too weak.

5. A note. At the end of the paper, we point out why we do not use another different formulation of (2). Denote the robustly optimal solution set of (1) by \( \text{OPT}(l) \). If we replace \( F(l) \) with \( \text{OPT}(l) \) in (2), then we obtain a new model as follows:

\[
\begin{align*}
\text{max} & \quad l \\
\text{s.t.} & \quad x \in \text{OPT}(l) \\
& \quad x \in S = \{ x \in \mathbb{R}^n | Ex \leq f \} \\
& \quad l \geq \bar{l}, l \geq 0.
\end{align*}
\] (19)

We take the case that perturbation only occurs in the right-hand-side vector \( b \) as an example. According to the strong duality theory of linear programming, (19) is equivalent to the following bilinear problem:

\[
\begin{align*}
\text{max} & \quad l \\
\text{s.t.} & \quad (A_0 \ I) \begin{pmatrix} x \\ t \end{pmatrix} = b_0 - lv \\
& \quad A_0^T y + s = c \\
& \quad c^T x = (b_0 - lv)^T y \\
& \quad Ex \leq f \\
& \quad l \geq \bar{l} \\
& \quad x \in \mathbb{R}_+^n, l \in \mathbb{R}_+, t \in \mathbb{R}_+^m, s \in \mathbb{R}_+^{n+m}, y \in \mathbb{R}_m.
\end{align*}
\] (20)

When \( l \) is fixed, all constraints are linear. Therefore, whether \( l \) is feasible for (20) can be determined in polynomial time. We assume the problem above is feasible. Then is the feasible region of \( l \) connected? In other words, when \( l_1, l_2 \) are both feasible for (20) with \( \bar{l} \leq l_1 < l_2 \), is \( l \) feasible for any \( l \in (l_1, l_2) \)? If yes, we can use
the binary search algorithm. Unfortunately, $OPT(l)$ does not intersect the polytope \[ \{ x \in \mathbb{R}^n \mid Ex \leq f \} \] necessarily for \( l \in [l_1, l_2] \). Thus this model is meaningless and the binary search algorithm is invalid here. We present the following example to illustrate it.

**Example 5.**

\[
\begin{align*}
\text{min} & \quad -30x_1 - 60x_2 - 20x_3 \\
\text{s.t.} & \quad 3x_1 + 4x_2 + x_3 \leq 20 \\
& \quad x_1 + 3x_2 + 2x_3 \leq 10 \\
& \quad x_i \geq 0, \ i = 1, 2, 3.
\end{align*}
\]

The optimal solution is \((4, 2, 0)^T\). And we let the perturbation direction vector of \( b \) be \((2, 0)^T\) and \( S = \{ x \mid x_3 = 0 \} \).

For \( l = 3 \), \((0.4, 3.2, 0)^T\) is feasible for (20) and is optimal to (1) which is the robust counterpart of the problem above. For \( l = 6 \), (20) is not feasible and there is no optimal solution to (1) with the third entry being 0. For \( l = 10 \), \((0, 0, 0)^T\) is feasible for (20) and is optimal to (1), and now the feasible set of (1) is a singleton set including \((0, 0, 0)^T\) only.

Thus \( l = 3 \) and \( l = 10 \) are both feasible for (19) while \( l = 6 \) is not.

Particularly, when \( S = \{ x^* \} \) where \( x^* \) is an optimal solution of (1), the operation that replacing \( F(l) \) with \( OPT(l) \) does not make any difference.

6. Conclusions. In this study, we introduce a new concept—robust sensitivity analysis, which is really different from the classical sensitivity analysis and some new variants of robust optimization. We have established a unified model based on this concept and have provided analyses in three perturbation cases. All the programs in our study can be equivalently reformulated into polynomial-time solvable models except for the ones where more than one row of the constraint matrix is perturbed and the feasible solution is not fixed (i.e., the polytope \( S \) is not a singleton set). For those, we have developed a binary search algorithm to obtain the maximal perturbation radius and it is still open whether there exist equivalent polynomial-time solvable models with regard to them. Besides, this concept can be also used for the convex quadratically constrained quadratical programming (CQCQP) problem, which has been considered in another paper. Also, the robust sensitivity analysis for the semidefinite programming (SDP) problem is a potential research direction.

Acknowledgments. The authors wish to thank the editor and the two anonymous referees for their insightful constructive suggestions to improve the quality of this paper.

REFERENCES

[1] A. Ben-Tal, D. D. Hertog, A. D. Waegenaere, B. Melenberg and G. Rennen, Robust solutions of optimization problems affected by uncertain probabilities, Management Science, 59 (2013), Pages iv–527.
[2] A. Ben-Tal and A. Nemirovski, Lectures on Modern Convex Optimization: Analysis, Algorithms and Engineering Applications, Society for Industrial and Applied Mathematics, 2001.
[3] D. Bertsimas and D. B. Brown, Constructing uncertainty sets for robust linear optimization, Operations Research, 57 (2009), 1483–1495.
[4] D. Bertsimas, V. Gupta and N. Kallus, Data-driven robust optimization, Mathematical Programming, 167 (2018), 235–292.
[5] D. Bertsimas and M. Sim, The price of robustness, Operations Research, 52 (2004), 35–53.
[6] F. Chandra, D. F. Gayme, L. Chen and J. C. Doyle, Robustness, optimization, and architectures, European Journal of Control, 17 (2011), 472–482.
[7] G. E. Dullerud and F. G. Paganini, A Course in Robust Control Theory—A Convex Approach, Springer-Verlag, New York, 2000.
[8] M. Frank and P. Wolfe, An algorithm for quadratic programming, Naval Research Logistics Quarterly, 3 (1956), 95–110.
[9] S. I. Gass, Linear Programming: Methods and Applications, Corrected reprint of the 1995 fifth edition. Dover Publications, Inc., Mineola, NY, 2003.
[10] J. Gorski, F. Pfeuffer and K. Klamroth, Biconvex sets and optimization with biconvex functions: A survey and extensions, Mathematical Methods of Operations Research, 66 (2007), 373–407.
[11] C.-L. Hwang and A. S. Masud, Multiple Objective Decision Making—Methods and Applications: A State-of-the-Art Survey, Springer-Verlag, Berlin, 1979.
[12] A. Marandi, A. Ben-Tal, D. D. Hertog and B. Melenberg, Extending the scope of robust quadratic optimization, Available on Optimization Online, 2017.
[13] S. Schaible, Parameter-free convex equivalent and dual programs of fractional programming problems, Zeitschrift für Operations Research, 18 (1974), 187–196.
[14] A. L. Soyster, Convex programming with set-inclusive constraints and applications to inexact linear programming, Operations Research, 22 (1974), 892–898.
[15] G. Xu and S. Burer, Robust sensitivity analysis of the optimal value of linear programming, Optimization Methods and Software, 32 (2017), 1187–1205.
[16] K. Zhou, J. C. Doyle and K. Glover, Robust and Optimal Control, Prentice Hall Englewood Cliffs, NJ, 1996.

Received September 2018; 1st revision December 2018; 2nd revision January 2019.

E-mail address: gaort17@mails.tsinghua.edu.cn
E-mail address: wxing@math.tsinghua.edu.cn