

Robust Sensitivity Analysis for Linear Programming with Ellipsoidal Perturbation

Ruotian Gao and Wenxun Xing
Department of Mathematical Sciences
Tsinghua University, Beijing, China, 100084

September 27, 2017

Abstract

Given an originally robust optimal decision and allowing perturbation parameters of the linear programming problem to run through a maximum uncertainty set controlled by a variable of perturbation radius, we do robust sensitivity analysis for the robust linear programming problem in two scenarios. One is to keep the original decision still robust optimal, the other is to ensure some properties of the original decision preserved in the robust optimal solution set. In each scenario, we do analyses in three cases with different perturbation styles. All models in our study are formulated into either linear programs or second-order conic programs except for some cases considering more than one row perturbation in the constraint matrix. For those, we develop a binary search algorithm.

Keywords: Linear programming, sensitivity analysis, robust optimization, second-order cone programming.

1 Introduction

In this study, the perturbed linear programming problem is as follows:

$$\begin{aligned} \min \quad & c_0^T x \\ \text{s.t.} \quad & Ax \leq b, \forall (A, b) \in U(l), \\ & x \geq 0, \end{aligned} \tag{1.1}$$

where $x \in \mathbb{R}^n$ is a variable, $c_0 \in \mathbb{R}^n$ is fixed and $(A, b) \in \mathbb{R}^{m \times n} \times \mathbb{R}^m$ is in an ellipsoidal uncertainty set $U(l)$ determined by a variable l of perturbation radius (radii) that is nonnegative. Without loss of generality, we do not consider uncertainty in the coefficients of the objective function since we can equivalently reformulate the problem by introducing a new variable t , replacing the objective with $\min t$ and adding a new constraint $c_0^T x \leq t$. In this study, we mainly discuss three perturbation styles that the right-hand-side vector is perturbed holistically, the constraint matrix is perturbed row-wisely and holistically, respectively. Here we will not consider the simultaneous perturbation of the constraint matrix and the right-hand-side vector since we can reform it into

$$\begin{aligned}
& \min \quad c_0^T x \\
& \text{s.t.} \quad \begin{pmatrix} A & b \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \leq 0, \quad \forall (A, b) \in U(l), \\
& \quad \quad y = -1, \\
& \quad \quad x \geq 0,
\end{aligned} \tag{1.2}$$

where perturbation only appears in the constraint matrix. The uncertainty set $U(l)$ in those three styles is given as below.

Case 1: right-hand-side vector perturbed holistically

$$U(l) = \{A_0\} \times U_b = \{A_0\} \times \left\{ b \in \mathbb{R}^m \mid b = b_0 + \sum_{j=1}^t \beta_j v_j, \|\beta\| \leq l \right\}. \tag{1.3}$$

Case 2: constraint matrix perturbed row-wisely, where $l = (l^1, l^2, \dots, l^m)^T$,

$$\begin{aligned}
U(l) &= U_{A1} \times \{b_0\} \\
&= \left\{ A \in \mathbb{R}^{m \times n} \mid A^i = A_0^i + \left(\sum_{j=1}^{n_i} \alpha_j^i u_j^i \right)^T, \|\alpha^i\| \leq l^i, i = 1, \dots, m \right\} \times \{b_0\}.
\end{aligned} \tag{1.4}$$

Case 3: constraint matrix perturbed holistically

$$U(l) = U_{A2} \times \{b_0\} = \left\{ A \in \mathbb{R}^{m \times n} \mid A = A_0 + \sum_{j=1}^s \gamma_j A_j, \|\gamma\| \leq l \right\} \times \{b_0\}. \tag{1.5}$$

Here $\beta = (\beta_1, \dots, \beta_t)^T$, $\alpha^i = (\alpha_1^i, \dots, \alpha_{n_i}^i)^T$, $i = 1, \dots, m$ and $\gamma = (\gamma_1, \dots, \gamma_s)^T$ are perturbation parameters taking values arbitrarily in some balls. A_0^i is the i -th row of A_0 and A^i is the i -th row of A^i . $v_j, j = 1, \dots, t; u_j^i, j = 1, \dots, n_i, i = 1, \dots, m$ are perturbation direction vectors and $A_j, j = 1, \dots, s$ are perturbation direction matrices. All of them are given by experience and estimation with no distributional assumption on themselves, since we do not have numerous data to obtain a probabilistic description of the uncertainty set, which is different from stochastic programming referred in [7], [8] and [11]. Notice that l is a vector in the second case and is a number in the other two cases. In case of $U(l) = \{A_0\} \times \{b_0\}$ when $l = 0$, (1.1) is indeed the original linear programming problem which is supposed to be solvable.

In classical sensitivity analysis for the linear programming problem, the ranges of the objective coefficients and the ranges of the right-hand-side vector are analyzed to keep the optimal solution unchanged or the optimal basis unchanged, respectively, which can be easily found in books about linear programming. In both cases, the ranges of the parameters are given in a "box" form. However, an optimal solution to one linear program with the parameters fixed may be infeasible for another realization of the linear program in the classical sensitivity analysis. To get a solution capable of coping with all possible realizations of the parameters, the robust optimization model is a promising choice.

When l is fixed, (1.1) is the robust optimization counterpart of linear programming. Robust optimization which came from the robust control community (refer to [9] and [14]) dates back to the work in [12], and has developed rapidly in the past four decades, see e.g., [6], [2] and the references therein. The form of uncertainty sets, the performance assessment of robust optimal solutions and the computational complexity of the models are three main topics considered in the robust optimization modeling. Based on the basic concept of the robust optimization, the uncertainty set(s) should contain all the observed data. So the easiest one is an interval by providing the minimum and maximum of the data for each individual parameter. The uncertainty sets like ellipsoids or polytopes are then used to describe those data. For the interval sets, [5] found the traditional robust optimization model is rather conservative for 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, [4] established convex uncertainty sets like polytopes and ellipsoids by using distortion measures

to evaluate the feasibility. All models in [4] and [5] preserve computational tractability which are polynomially solvable. But any robust optimal solution of the models in [4] and [5] may not be feasible with a probability for the original robust problems, which somehow does not satisfy the concept of robust optimization.

Now we consider the robust optimization in a different way. When the uncertainty set is described by the ellipsoidal form, the robust linear programming is polynomially computable for any given l , see [1]. Then for any robust optimal solution selected with a given l , is this given solution still robust optimal for a larger data perturbation? Under the assumption of the uncertainty set of ellipsoids, the radius of any ellipsoid is regarded as a variable and the maximum radii should be calculated to obtain the maximum data perturbation. This is our robust sensitivity analysis concept.

More concretely, two scenarios are considered in this study. One is to keep the originally robust optimal solution x^* still robustly optimal, the other is to keep 0 entries of x^* preserved in a robust optimal solution set. These two scenarios really make sense in decision-making of practical production process, investment and other fields. For example, we have got the optimal strategy for the current stage. In the coming stage, we just have old information and we have to make a decision before parameters are exactly known. Of course, it would be the best if the robust optimal solution is still x^* since we need not change the decision at all, which corresponds to the first scenario. Or at least, it is an acceptable decision that the items not considered in the current stage will not be considered in the coming stage, avoiding new setup costs, which corresponds to the second scenario. For simplicity, we call the two scenarios as "optimal solution unchanged" and "0 entries unchanged" respectively.

In our study, two main models are provided associated with the two scenarios. Model 1 for the first scenario is:

$$\begin{aligned}
 & \max \quad l \\
 & \text{s.t.} \quad x^* \in \mathcal{OPT}_1(l), \\
 & \quad \quad l \geq 0,
 \end{aligned} \tag{1.6}$$

where $\mathcal{OPT}_1(l)$ is the optimal solution set of (1.1). Model 2 for the second scenario is:

$$\begin{aligned}
& \max \quad l \\
& \text{s.t.} \quad \mathcal{OPT}_2(l) \neq \emptyset, \\
& \quad \quad \quad l \geq 0,
\end{aligned} \tag{1.7}$$

where $\mathcal{OPT}_2(l)$ is the optimal solution set of the following perturbed problem derived from (1.1) with an additional constraint $x_j = 0$, $j \in P$, where $P = \{j \mid x_j^* = 0\}$ is a given index set determined by the given decision x^* ,

$$\begin{aligned}
& \min \quad c_0^T x \\
& \text{s.t.} \quad Ax \leq b, \quad \forall (A, b) \in U(l), \\
& \quad \quad \quad x_j = 0, \quad j \in P, \\
& \quad \quad \quad x \geq 0.
\end{aligned} \tag{1.8}$$

Model 1 is to find the maximum l of radius (radii) to keep the originally optimal solution x^* still robust optimal to (1.1). Model 2 is to find the maximum l of radius (radii) to ensure the robust optimal solution set of (1.8) nonempty.

[15] presented copositive programs to state the best-case and worst-case optimal values when the coefficients in the objective function and the right-hand side are perturbed. And they developed tight, tractable conic relaxations to provide the lower and upper bounds. They called their work as "robust sensitivity analysis of the optimal value of linear programming". However, their idea is quite different from ours. Their uncertainty set is fixed rather than variable, their feasible solution set is not suitable for all possible realizations of the parameters and perturbation only in objective and right-hand side is allowed, while the constraint matrix is not considered.

Our main contributions are stated as follows. First, to the best of our knowledge, this robust sensitivity analysis concept is first brought up by us and can be applied to evaluate the robustness of a pre-decision. Second, we establish models corresponding to two scenarios, which shows the possibility of applying our concept. The models are either linear or bi-convex programs for the three perturbation cases. The bi-convex programs can be equivalently formulated into second-order conic programs when only one row perturbation is considered. While

for those models considering more than one row perturbation, we haven't found equivalent polynomial ones and thus provide a binary search algorithm, which triggers our interest in the bi-convex optimization problems.

Now we provide some notations needed in this paper. \mathbb{R}^n denotes the set of n dimensional real vectors, \mathbb{R}_+^n denotes the set of n dimensional nonnegative real vectors. 2-norm is used in this paper, namely, $\|x\|_2 = (\sum_{i=1}^n x_i^2)^{1/2}$, where $x \in \mathbb{R}^n$, and causing no confusion, we omit the subscript 2 afterwards. And $x^T y = \sum_{i=1}^n x_i y_i$, where $x, y \in \mathbb{R}^n$. In addition, all the vectors mentioned in this paper are column vectors.

The rest of the paper is organized as follows. In Section 2 and Section 3, we do robust analysis in two scenarios respectively, and in each scenario, three cases are considered. In Section 4, numerical experiment is showed. In addition, two notes and the conclusions are given in Section 5 and Section 6.

2 Optimal Solution Unchanged

This section gives analysis of the first scenario (1.6) in the three cases. In our assumption, x^* is supposed to be a robust optimal solution for a given \bar{l} . We will get the maximum radius (radii) of l , no smaller than \bar{l} , such that x^* is still robust optimal. For simplicity, let $\bar{l} = 0$ in the following consideration.

2.1 b_0 Perturbed Holistically

Case 1 is considered in this subsection and the perturbed problem is

$$\begin{aligned} \min \quad & c_0^T x \\ \text{s.t.} \quad & A_0 x \leq b_0 + \sum_{j=1}^t \beta_j v_j, \quad \forall \|\beta\| \leq l, \\ & x \geq 0. \end{aligned} \tag{2.1}$$

where $\beta = (\beta_1, \dots, \beta_t)^T \in \mathbb{R}^t$ runs through the ball centered at the origin with a radius l . Let $v_i \in \mathbb{R}^t$ consists of all the i -th entries of $v_j, j = 1, \dots, t$. Then

$$\begin{aligned}
& x \text{ is feasible for (2.1)} \\
& \iff (A_0x - b_0)_i \leq \min_{\|\beta\| \leq l} \beta^T v_i, \quad i = 1, \dots, m, \text{ and } x \geq 0 \\
& \iff (A_0x - b_0)_i \leq -l\|v_i\|, \quad i = 1, \dots, m, \text{ and } x \geq 0.
\end{aligned}$$

Denote $v = (\|v_1\|, \dots, \|v_m\|)^T \in \mathbb{R}_+^m$. (2.1) is equivalent to

$$\begin{aligned}
& \min \quad c_0^T x \\
& \text{s.t.} \quad A_0x \leq b_0 - lv, \\
& \quad \quad x \geq 0.
\end{aligned} \tag{2.2}$$

Now in Model 1, $\mathcal{OPT}_1(l)$ stands for the optimal solution set of (2.2).

Lemma 1. *For x^* , feasibility is enough to promise optimality.*

Proof: Since l and v are both nonnegative, we have $A_0x \leq b_0$ for any x feasible for (2.2), thus $x \in \mathcal{F}_0$. Then we have $c_0^T x^* \leq c_0^T x$ because x^* is originally optimal. So once x^* is feasible for (2.2), it is optimal. \square

Then (1.6) of Model 1 is formulated as

$$\begin{aligned}
& \max \quad l \\
& \text{s.t.} \quad A_0x^* \leq b_0 - lv, \\
& \quad \quad l \geq 0.
\end{aligned} \tag{2.3}$$

Remark 1. *The feasible region of l in (2.3) is an interval whose left end point is zero.*

The following results are obvious by Lemma 1 and (2.3).

Theorem 1. *For (2.3),*

- (i) *if there exists i such that $(A_0x^* - b_0)_i = 0$ and $\|v_i\| > 0$, the maximum radius $l^* = 0$;*
- (ii) *if (i) does not hold for any $i = 1, \dots, m$, denote $J = \{i \mid \|v_i\| > 0\}$. If J is empty, $l^* = \infty$, otherwise $l^* = \min_{i \in J} (b_0 - A_0x^*)_i / \|v_i\| > 0$.*

In case of (i) above, the constraint is called active. Although β cannot take values arbitrarily in a ball with a positive radius for an active constraint, it can vary in certain directions. For some

i which satisfies $(A_0x^* - b_0)_i = 0$ and $\|v_i\| > 0$, the inequality $\beta^T v_i \geq 0$ must hold, which means the angle between β and v_i is no larger than $\pi/2$. Then $D_\beta = \{\beta \mid \beta^T v_i \geq 0, i \in I_1\}$ stands for an alternative direction set of β , where $I_1 = \{i \mid (A_0x^* - b_0)_i = 0 \text{ and } \|v_i\| > 0\}$. So we can only select β in the set D_β if it is nonempty. Denote $I_2 = \{i \mid (A_0x^* - b_0)_i < 0 \text{ and } \|v_i\| > 0\}$. If I_2 is nonempty, the maximum norm l_1^* of β is $\min_{i \in I_2} (b_0 - A_0x^*)_i / \|v_i\|$ when β is selected arbitrarily in D_β , otherwise $l_1^* = +\infty$. Now the region where β can vary arbitrarily is a part of a ball rather than a whole ball. Obviously, the problem cannot be perturbed at all when D_β is empty.

The result that any linear programming problem can get its optimum at boundary points implies that (i) happens generally. In addition to treating case (i) like above, we can introduce a tolerance $\delta > 0$ to $(b_0)_i$ and the constraint becomes $(A_0x - b_0)_i \leq \delta$. Now the parameters in the i -th constraint can be perturbed and the maximum perturbation radius is $\delta / \|v_i\| > 0$. Of course, a bigger tolerance we permit, a bigger perturbation radius we will obtain. In practical production, a tolerance is reasonable since the manufacturer had better replenish one resource when it is used up exactly in the current stage. Here δ can be regarded as the extra supplement quantity of this resource.

2.2 A_0 Perturbed Row-wisely

Case 2 is considered in this subsection and the perturbed problem is

$$\begin{aligned}
\min \quad & c_0^T x \\
\text{s.t.} \quad & (A_0^i + (\sum_{j=1}^{n_i} \alpha_j^i u_j^i)^T) x \leq b_0^i, \quad \forall \|\alpha^i\| \leq l^i, \quad i = 1, \dots, m, \\
& x \geq 0,
\end{aligned} \tag{2.4}$$

where $\alpha^i = (\alpha_1^i, \dots, \alpha_{n_i}^i)^T \in \mathbb{R}^{n_i}$ runs through the ball centered at the origin with radius l^i . Here A_0^i is the i -th row of A_0 and b_0^i is the i -th entry of b_0 . It is equivalent to

$$\begin{aligned}
\min \quad & c_0^T x \\
\text{s.t.} \quad & A_0^i x + l^i \|H_i x\| \leq b_0^i, \quad i = 1, \dots, m, \\
& x \geq 0,
\end{aligned} \tag{2.5}$$

where $H_i = (u_1^i, \dots, u_{n_i}^i)^T \in \mathbb{R}^{n_i \times n}$. Now $\mathcal{OPT}_1(l)$ stands for the optimal solution set of (2.5). Similarly, feasibility promises optimality. Then (1.6) of Model 1 is formulated as

$$\begin{aligned} \max \quad & l \\ \text{s.t.} \quad & A_0^i x^* + l^i \|H_i x^*\| \leq b_0^i, \quad i = 1, \dots, m, \\ & l \geq 0. \end{aligned} \tag{2.6}$$

Here $l = (l^1, \dots, l^m)^T$, thus this is a multi-objective program. Since l^i , $i = 1, \dots, m$ are independent, maximizing l is equivalent to maximizing all l^i , $i = 1, \dots, m$. Then we can divide (2.6) into m subproblems and consider the following subproblem for each constraint i ,

$$\begin{aligned} \max \quad & l^i \\ \text{s.t.} \quad & A_0^i x^* + l^i \|H_i x^*\| \leq b_0^i, \\ & l^i \geq 0. \end{aligned} \tag{2.7}$$

Theorem 2. For (2.7), if $\|H_i x^*\| = 0$, the maximum radius $l^{i*} = +\infty$, otherwise $l^{i*} = (b_0^i - A_0^i x^*) / \|H_i x^*\|$.

It is lucky to find that $l^* = (l^{1*}, \dots, l^{m*})^T$ is an absolute optimal solution to the multi-objective programming problem (2.6), namely, for any feasible solution l , we have $l^* \geq l$. Similarly, we can introduce a tolerance $\delta > 0$ for those active constraints to avoid unperturbational cases.

2.3 A_0 Perturbed Holistically

Case 3 is considered in this subsection and the perturbed problem is

$$\begin{aligned} \min \quad & c_0^T x \\ \text{s.t.} \quad & (A_0 + \sum_{j=1}^s \gamma_j A_j) x \leq b_0, \quad \forall \|\gamma\| \leq l, \\ & x \geq 0, \end{aligned} \tag{2.8}$$

where $\gamma = (\gamma_1, \dots, \gamma_s)^T \in \mathbb{R}^s$ runs through the ball centered at the origin with radius l . Denote $U_i = (A_1^{iT}, \dots, A_s^{iT})^T \in \mathbb{R}^{s \times n}$, $i = 1, \dots, m$. Then (2.8) is equivalent to

$$\begin{aligned} \min \quad & c_0^T x \\ \text{s.t.} \quad & A_0^i x + l \|U_i x\| \leq b_0^i, \quad i = 1, \dots, m, \\ & x \geq 0. \end{aligned} \tag{2.9}$$

Similarly, (1.6) of Model 1 is formulated as

$$\begin{aligned} \max \quad & l \\ \text{s.t.} \quad & A_0^i x^* + l \|U_i x^*\| \leq b_0^i, \quad i = 1, \dots, m, \\ & l \geq 0. \end{aligned} \tag{2.10}$$

Theorem 3. For (2.10), if $\|U_i x^*\| = 0, \forall i = 1, \dots, m$, the maximum radius $l^* = +\infty$, otherwise $l^* = \min_{i \in K} (b_0^i - A_0^i x^*) / \|U_i x^*\|$, where $K = \{i \mid \|U_i x^*\| > 0\}$.

Similarly, we can introduce a tolerance $\delta > 0$ for those active constraints to avoid unperturbational cases.

3 0 Entries Unchanged

This section gives analysis of the second scenario (1.7) in the three cases. When l is a vector, Model 2 is a multi-objective programming. Generally, there does not necessarily exist an absolute optimal solution to a multi-objective programming, refer to [10]. See the following example.

Example 1.

$$\begin{aligned} \min \quad & -2x_1 - x_2 + 2x_3 \\ \text{s.t.} \quad & 3x_1 + 4x_2 + x_3 \leq 2, \\ & -x_1 - 3x_2 - 2x_3 \leq -1, \\ & x_i \geq 0, \quad i = 1, \dots, 3. \end{aligned} \tag{3.1}$$

An optimal solution $x^* = (0.4, 0.2, 0)^T$ of the example above is selected and $P = \{3\}$ here. The perturbation direction vectors for the first row are $u_1^1 = (0.1, 0.1, 0.2)^T$ and $u_2^1 = (0.3, 0.2, -0.1)^T$, and the one for the second row is $u_1^2 = (-0.1, -0.2, 0.1)^T$, then

$$H_1 = \begin{pmatrix} 0.1 & 0.1 & 0.2 \\ 0.3 & 0.2 & -0.1 \end{pmatrix}, H_2 = \begin{pmatrix} -0.1 & -0.2 & 0.1 \end{pmatrix}. \quad (3.2)$$

Then (1.7) is formulated as follows:

$$\begin{aligned} \max \quad & l \\ \text{s.t.} \quad & 3x_1 + 4x_2 + x_3 + l^1 \|H_1 x\| \leq 2, \\ & -x_1 - 3x_2 - 2x_3 + l^2 \|H_2 x\| \leq -1, \\ & x_3 = 0, \\ & l \geq 0, \quad x_i \geq 0, \quad i = 1, 2, 3, \end{aligned} \quad (3.3)$$

where $l = (l^1, l^2)^T$. First, we only perturb the first row, namely let $l^2 = 0$. Actually, (3.3) with $l^2 = 0$ can be reformed into a second-order cone program which will be proved later, and the maximum l^1 is 8.9443 leading to a feasible solution $l = (8.9443, 0)^T$ of (3.3). Similarly, the maximum l^2 is 5.0000 when l^1 is fixed to be 0 so $l = (0, 5.0000)^T$ is also feasible for (3.3). Henceforth, if there exists an absolute optimal solution l , we must have $l \geq (8.9443, 0)^T$ and $l \geq (0, 5.0000)^T$. However, even the most promising choice $l = (8.9443, 5.0000)^T$ fails to be feasible.

Therefore, we will provide two indirect models below to turn a multi-objective programming into a single-objective programming and will discuss them in detail in subsequent subsections.

- Preference model:

$$\begin{aligned} \max \quad & l^k \\ \text{s.t.} \quad & \mathcal{OPT}_2(l) \neq \emptyset, \\ & l = l^k g, \\ & l^k \geq 0, \end{aligned} \quad (3.4)$$

where k is a selected index, and $g \in \mathbb{R}_+^m$ is a given vector with $g_k = 1$. $g_i, i \neq k$ represents

the weight of the i -th row compared with the k -th row . For example, if we hope the i -th row has a greater space to be perturbed than the k -th row, we can let $g_i > 1$. The method is used when we have a preference on the rows.

- Maxmin model:

$$\begin{aligned}
 & \max \min_{i=1, \dots, m} l^i \\
 & \text{s.t. } \mathcal{OPT}_2(l) \neq \emptyset, \\
 & \quad l \geq 0.
 \end{aligned} \tag{3.5}$$

The method is used when we expect to make the minimum radius as large as possible.

3.1 b_0 Perturbed Holistically

With almost the same argument of getting (2.2), the perturbed problem is

$$\begin{aligned}
 & \min \quad c_0^T x \\
 & \text{s.t. } \quad A_0 x \leq b_0 - lv, \\
 & \quad x_j = 0, \quad j \in P, \\
 & \quad x \geq 0.
 \end{aligned} \tag{3.6}$$

Now in Model 2, $\mathcal{OPT}_2(l)$ stands for the optimal solution set of (3.6). Since the problem is attainable if feasible, (1.7) of Model 2 is formulated as the following linear program:

$$\begin{aligned}
 & \max \quad l \\
 & \text{s.t. } \quad A_0 x \leq b_0 - lv, \\
 & \quad x_j = 0, \quad j \in P, \\
 & \quad x \geq 0, \quad l \geq 0.
 \end{aligned} \tag{3.7}$$

3.2 A_0 Perturbed Row-wisely

With almost the same argument of getting (2.5), the perturbed problem is

$$\begin{aligned}
\min \quad & c_0^T x \\
\text{s.t.} \quad & A_0^i x + l^i \|H_i x\| \leq b_0^i, \quad i = 1, \dots, m, \\
& x_j = 0, \quad j \in P, \\
& x \geq 0,
\end{aligned} \tag{3.8}$$

Then (1.7) of Model 2 is formulated as

$$\begin{aligned}
\max \quad & l \\
\text{s.t.} \quad & A_0^i x + l^i \|H_i x\| \leq b_0^i, \quad i = 1, \dots, m, \\
& x_j = 0, \quad j \in P, \\
& x \geq 0, \quad l \geq 0,
\end{aligned} \tag{3.9}$$

where $l = (l^1, \dots, l^m)^T \in \mathbb{R}^m$. (3.9) cannot be solved easily owing to the cross terms $l^i \|H_i x\|$, $i = 1, \dots, m$. To analyze this problem, we need an assumption here.

Assumption 1. *The feasible region of the problem (3.8) with $l = 0$ is bounded and has a strict interior point. Here a strict interior point means there exists a feasible solution denoted as \bar{x} such that $A_0^i \bar{x} < b_0^i$, $i = 1, \dots, m$.*

Obviously, it is required that the primal unperturbed problem (1.1) with $l = 0$ has an interior feasible solution under this assumption.

As stated before, (3.9) may not have an absolute optimal solution, thus the first indirect model (3.4) is formulated as

$$\begin{aligned}
\max \quad & l^k \\
\text{s.t.} \quad & A_0^i x + g_i l^k \|H_i x\| \leq b_0^i, \quad i = 1, \dots, m, \\
& x_j = 0, \quad j \in P, \\
& x \geq 0, \quad l^k \geq 0.
\end{aligned} \tag{3.10}$$

Here $k \in \{1, \dots, m\}$ and $g \in \mathbb{R}_+^m$ is given with $g_k = 1$. Denote $Supp = \{i \mid g_i > 0\}$.

Proposition 1. *The optimal value of (3.10) is strictly larger than 0.*

Proof: By Assumption 1, there exists \bar{x} such that $A_0^i \bar{x} < b_0^i$, $i = 1, \dots, m$, $\bar{x}_j = 0$, $j \in P$ and $\bar{x} \geq 0$, denote $W = \{i \mid \|H_i \bar{x}\| > 0\} \subset \{1, \dots, m\}$. If $Supp \cap W \neq \emptyset$, let

$$l = \min \{(b_0^i - A_0^i \bar{x}) / (g_i \|H_i \bar{x}\|), i \in Supp \cap W\} > 0, \quad (3.11)$$

then l is feasible for (3.10) and the optimal value is larger than 0. If $Supp \cap W = \emptyset$, (\bar{x}, l) is feasible for (3.10) for any $l > 0$, then the optimal value is infinity which is larger than 0. Therefore, the problem can really be perturbed. \square

In order to determine when the maximum perturbation radius of (3.10) is finite, we have the following theorem.

Theorem 4. *The optimal value of (3.10) is finite if and only if the following problem has a positive optimal value.*

$$\begin{aligned} \min \quad & \|Hx\| \\ \text{s.t.} \quad & A_0 x \leq b_0, \\ & x_j = 0, j \in P, \\ & x \geq 0, \end{aligned} \quad (3.12)$$

where $H \in \mathbb{R}^{(\sum_{i \in Supp} n_i) \times n}$ consists of H_i , $i \in Supp$ in turn.

Proof: On one hand, if the optimal value of (3.12) is 0, then there exists an \tilde{x} such that $A_0^i \tilde{x} \leq b_0^i$, $i = 1, \dots, m$, $\tilde{x}_j = 0$, $j \in P$, $\tilde{x} \geq 0$ and $H_i \tilde{x} = 0$, $i \in Supp$. Then (\tilde{x}, l) is feasible for (3.10) for any $l > 0$ leading to an infinite optimal value of (3.10), which causes a contradiction.

On the other hand, if the optimal value of (3.10) is infinite, let $\{l^{(p)}, p = 1, \dots\}$ be a positive series with $l^{(p)} \rightarrow +\infty$. Then for any $l^{(p)}$, there exists $x^{(p)}$ such that $x_j^{(p)} = 0$, $j \in P$, $x^{(p)} \geq 0$, $A_0^i x^{(p)} \leq b_0^i$, $i \notin Supp$ and $\|H_i x^{(p)}\| \leq (b_0^i - A_0^i x^{(p)}) / g_i l^{(p)}$, $i \in Supp$. By the boundedness of Assumption 1 which is equivalent to that the feasible region of (3.10) with $l^k = 0$ is bounded, the feasible set related to x of (3.10) is bounded and closed. Then there exists $M > 0$ such that $b_0^i - A_0^i x \leq M$, $i = 1, \dots, m$ hold for any feasible solution x of (3.10). Now $\|H_i x^{(p)}\| \leq M / g_i l^{(p)}$, $i \in Supp$. Since the feasible region of (3.10) is bounded and

closed, we can select a convergent subsequence from $\{x^{(p)}, p = 1, \dots\}$, which is also denoted as $\{x^{(p)}, p = 1, \dots\}$ for convenience with $\|H_i x^{(p)}\| \rightarrow 0, i \in \text{Supp}$ and $x^{(p)} \rightarrow x^0$. It is obvious that x^0 is also feasible for (3.10) with $\|H_i x^0\| = 0, i \in \text{Supp}$, which leads to a contradiction to a positive optimal value of (3.12). \square

(3.10) is a bi-convex optimization problem as it is a second-order cone program when l is fixed and is a linear program when x is fixed. Generally a bi-convex optimization problem is intractable. Next we consider a simple case $g_k = 1, g_i = 0, i \neq k$, where the perturbation only appears in the k -th row, which is actually polynomially computable.

Theorem 5. *Under Assumption 1, let $g_k = 1, g_i = 0, i \neq k$ in (3.10), then the optimal value of (3.10) is infinity if the optimal value of (3.12) is 0 and is finite otherwise. In case of finite optimal value, (3.10) and the following second-order cone program share the same optimal value.*

$$\begin{aligned}
& \max \quad p \\
& \text{s.t.} \quad \|H_k q\| \leq 1, \\
& \quad \quad p = b_0^k s - A_0^k q, \\
& \quad \quad A_0 q \leq b_0 s, \\
& \quad \quad q_j = 0, \quad j \in P, \\
& \quad \quad s \geq 0, \quad q \geq 0,
\end{aligned} \tag{3.13}$$

where $q = (q_1, \dots, q_n)^T \in \mathbb{R}^n, p \in \mathbb{R}, s \in \mathbb{R}$.

Proof: The first claim is obvious by Theorem 4. In case of the finite optimal value of (3.10), we have $\|H_k x\| > 0$ for any feasible solution x of (3.12), and then $t \geq \|H_k x\| > 0$ for any feasible solution x of (3.10) as the feasible set of (3.12) concludes that of (3.10). Thus the following

problem is well defined and is an equivalent reformulation of (3.10) with $g_k = 1, g_i = 0, i \neq k$.

$$\begin{aligned}
& \max && y/t \\
& \text{s.t.} && \|H_k x\| \leq t, \\
& && y = b_0^k - A_0^k x, \\
& && A_0 x \leq b_0, \\
& && x_j = 0, j \in P, \\
& && x \geq 0, t \geq 0.
\end{aligned} \tag{3.14}$$

Then we are to prove (3.14) is equivalent to (3.13).

On one hand, for any feasible solution (x, y, t) of (3.14), $t > 0$ holds. Thus, $(q, p, s) = (x/t, y/t, 1/t)$ is well defined and is feasible for (3.13). Therefore, the optimal value of (3.13) is no less than that of (3.14).

On the other hand, there exists a feasible solution (x^0, y^0, t^0) of (3.14) by the strict interior point assumption such that $y^0/t^0 > 0$. Then for any feasible solution (q, p, s) of (3.13), if $p = 0$, $y^0/t^0 > p = 0$ obviously holds; if $p > 0$, we claim that $s > 0$ holds. Otherwise, we have $\|H_k q\| \leq 1$, $p = -A_0^k q$, $A_0 q \leq 0, q_j = 0, j \in P$ and $q \geq 0$. And for any $d > 0$, $(x^{(d)}, y^{(d)}, t^{(d)}) = (dq + x^0, dp + y^0, d + t^0)$ is feasible for (3.14). Based on Assumption 1, y is also bounded due to the boundedness of x and the second constraint in (3.14). Since d can be taken arbitrarily large, we must have $q = 0$ and $p = 0$, which causes a contradiction to $p > 0$. Since $s > 0$, $(x, y, t) = (q/s, p/s, 1/s)$ is well defined and is feasible for (3.14) with $y/t = p$. As a consequence, the optimal value of (3.14) is no less than that of (3.13). Henceforth, the two problems share the same optimal value. \square

Theorem 5 presents a polynomially computable special case for Model 2, in which perturbation is considered in one row alone. However, we haven't found a polynomial formulation for a general g . Thus we provide a binary search algorithm to solve it. Obviously, the feasible region of l^k is an interval whose left end point is 0. Notice that once we obtain a strict interior point \bar{x} in Assumption 1, we can provide a positive lower bound for the optimal value. Then the algorithm is designed as follows:

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

Algorithm:

Step1: Solve (3.12), if the optimal value is 0, the maximum radius is $l^k = +\infty$ and goto Step4, otherwise goto Step2.

Step2: Obtain an interior point \bar{x}^1 and a positive lower bound lb for the maximum radius given by (3.11). And Let $p = 0$ and $q = 1/lb$.

Step3: Let $l^k = 2/(p + q)$ in (3.10). If (3.10) is feasible, let $q = 1/l^k$, otherwise let $p = 1/l^k$. If $q - p < \varepsilon$, goto Step4, otherwise continue.

Step4: Output the maximum radius l^k .

¹ To obtain an interior point, we just need to solve (3.8) with $l = 0$ and the objective function changed to 0. Since the built-in algorithm in MATLAB for linear programming is interior point algorithm, it will return the analytical center of the feasible region, which is certainly an interior point.

For the maxmin method (3.5), we get

$$\begin{aligned} \max \quad & l \\ \text{s.t.} \quad & A_0^i x + l^i \|H_i x\| \leq b_0^i, \quad i = 1, \dots, m, \\ & x_j = 0, \quad j \in P, \\ & l^i \geq l, \quad i = 1, \dots, m, \\ & x \geq 0, \quad l \geq 0. \end{aligned} \tag{3.15}$$

By the following Remark 2, we get the same algorithm above to solve (3.15).

Remark 2. When l is fixed, (3.15) is feasible if and only if (3.15) is feasible with $l^i = l$, $i = 1, \dots, m$.

Remark 3. When $g = (1, \dots, 1)^T$, (3.10) and (3.15) share the same optimal value.

3.3 A_0 Perturbed Holistically

With almost the same argument of getting (2.9), the perturbed problem is

$$\begin{aligned} \min \quad & c_0^T x \\ \text{s.t.} \quad & A_0^i x + l \|U_i x\| \leq b_0^i, \quad i = 1, \dots, m, \\ & x_j = 0, \quad j \in P, \\ & x \geq 0. \end{aligned} \tag{3.16}$$

Then (1.7) for Model 2 is formulated as

$$\begin{aligned}
& \max \quad l \\
& \text{s.t.} \quad A_0^i x + l \|U_i x\| \leq b_0^i, \quad i = 1, \dots, m, \\
& \quad \quad x_j = 0, \quad j \in P, \\
& \quad \quad x \geq 0, l \geq 0.
\end{aligned} \tag{3.17}$$

It is interesting that the problem above has almost the same formulation of (3.10) by fixing $g_i = 1, i = 1, 2, \dots, m$. When all the rows to be perturbed are in a row-wise fashion, the numbers of every row's perturbation direction vectors are the same, and all the perturbation radii are required to be equal, (3.10) of A_0 perturbed row-wisely is the same as (3.17) of A_0 perturbed holistically. Henceforth, the case of A_0 perturbed holistically is actually a specialization of A_0 perturbed row-wisely and all the results for A_0 perturbed row-wisely are true for A_0 perturbed holistically.

4 Numerical Experiments

We are to implement the algorithm for problem (3.10) with $g = (1, \dots, 1)^T$, this means we let the radii of all rows be the same. Actually, whether all entries of g are equal has no effect on the computational complexity of the problem. 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.

The cardinality of index set P is given as 30% of n , and \bar{x} is supposed to be a strict interior point with $\bar{x}_j = 0, j \in P$ and $\bar{x}_j = 1, j \notin P$. A_0 and c_0 are uniformly distributed in $[-10, 10]^{m \times n}$ and $[-10, 10]^n$ respectively. And $b_0 - A_0 \bar{x} | A_0$ is uniformly distributed in $[2, 10]^n$ to ensure $A_0 \bar{x} < b_0$ holds strictly. Here symbol $|A_0$ means conditional distribution. Besides, we impose a box constraint $x \in [-100, 100]^n$ to make the feasible region bounded. Under these initial settings, Assumption 1 is tenable. 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 $u_j^i, j = 1, \dots, n_i, i = 1, \dots, m$ 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 ϵ is 10^{-6} .

We will show the improvement from the lower bound (3.11) to the maximum radius, and

influence on CPU time of different combination of $(m, n, pert, scale)$, where m is the number of constraints, n is the dimensional number of variables, $pert$ is the perturbation ratio 25% or 50%, and $scale$ is the perturbation scale of u_j^i uniformly distributed in $[-0.5, 0.5]^n$ or $[-1, 1]^n$. Here we let numbers of perturbation vectors for all the rows be equal for simplicity. In every setting $(m, n, pert, scale)$, we do 10 random instances repeatedly. And *Ave* and *Std* stand for the average value and the standard deviation of the 10 experiments.

Table 1: Numerical Results

Instance ($m^1, n^2, pert^3, scale^4$)	Lower Bound		Maximum Radius		Improvement Ratio ⁵		CPU Time	
	Ave	Std	Ave	Std	Ave	Std	Ave	Std
(20,20,0.25,0.5)	0.4151	0.1551	1.1650	0.8647	2.3001	3.0673	15.8480	1.1491
(20,20,0.25,1)	0.2283	0.0969	0.7672	0.3066	2.6969	1.3452	15.6160	0.9467
(20,20,0.5,0.5)	0.3056	0.1443	1.0726	0.7539	2.8557	3.0440	17.0120	1.2904
(20,20,0.5,1)	0.1385	0.0818	0.4476	0.2327	2.6981	2.4415	18.3810	1.8375
(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	27.9500	1.7346
(20,100,0.5,0.5)	0.0800	0.0066	0.4237	0.2145	4.3204	2.7268	35.2770	1.7099
(20,100,0.5,1)	0.0435	0.0064	0.1897	0.2614	3.7842	7.3197	33.4340	1.9935
(80,100,0.25,0.5)	0.1381	0.0534	0.5341	0.2676	2.9914	1.9738	118.4410	5.8678
(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	2.4909	184.1960	9.4721
(80,100,0.5,1)	0.0450	0.0075	0.2548	0.0952	4.7366	2.3818	201.1790	20.2812

¹ number of constraints

² dimension of x

³ ratio of number of perturbation vectors for one row to $m + n$

⁴ scale of perturbation vectors

⁵ improvement ratio=(maximum radius-lower bound)/lower bound

For these numerical experiments in Table 1, infinite perturbation radius doesn't appear 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 number of perturbation vectors is, the more CPU time it costs whereas the scale of perturbation vectors basically has no influence on CPU time. As for the maximum radius, it is mainly affected by the scale of perturbation vectors. It is easily found that the maximum radius on level $[-0.5, 0.5]^n$ is approximately twice the maximum radius on level $[-1, 1]^n$. And the other three factors have less effect on it. And the improvement ratio is about between 2 and 6, which shows that a naive lower bound is too weak.

5 Notes

At the end of the paper, we make two notes on why we haven't studied on the optimal partition and another different formulation for Model 2.

First, the reason why we haven't studied on the optimal partition, which is focused on in traditional sensitivity analysis for the linear programming, is that the perturbation problems no longer remain linear such as (2.5). And there is no point in keeping the optimal partition unchanged, since the definition of optimal partition of linear programming and second-order cone programming is different, see [13]. Especially, for perturbation of b_0 , (2.2) is luckily a linear program and it can be regarded as a parametric linear programming problem, in which $-v$ is a given perturbation vector and l is a parameter causing the variation. This problem has been studied detailedly in [3], in which Ben-Tal and Nemirovski presented algorithms to compute all transition-points and showed that optimal partition remains constant between two consecutive transition-points. Therefore the maximum perturbation radius to keep the optimal partition of (2.2) unchanged can be obtained by the methods mentioned in that paper. In order to establish a unified analysis system for those three cases, we consider Model 2 to keep 0 entries remained, which is somehow like the invariance of the optimal partition.

Second, if we consider the condition $x_j = 0$, $j \in P$ out of $\mathcal{OPT}_2(l)$ for Model 2, we obtain a new model replacing (1.7) as follows:

$$\begin{aligned} \max \quad & l \\ \text{s.t.} \quad & x \in \mathcal{OPT}_1(l), \\ & x_j = 0, \quad j \in P, \\ & l \geq 0. \end{aligned} \tag{5.1}$$

For the general case reformulation (3.10) of (1.7), we use the binary search algorithm to get the maximum radii. When the objective l of (5.1) is one dimensional, which is similar to that of (3.10), can we use the binary search algorithm to solve it?

Here we consider the case that perturbation only occurs in the right-hand-side vector as an example. According to the strong duality theorem for the linear programming, (5.1) is

equivalent to the following bilinear problem,

$$\begin{aligned}
& \max \quad l \\
& \text{s.t.} \quad \begin{pmatrix} A_0 & I \end{pmatrix} \begin{pmatrix} x \\ t \end{pmatrix} = b_0 - lv, \\
& \quad \quad \begin{pmatrix} A_0^T \\ I \end{pmatrix} y + s = \begin{pmatrix} c_0 \\ 0 \end{pmatrix}, \\
& \quad \quad c_0^T x = (b_0 - lv)^T y, \\
& \quad \quad x_j = 0, \quad j \in P, \\
& \quad \quad x \geq 0, t \geq 0, s \geq 0, l \geq 0, y \in \mathbb{R}^m.
\end{aligned} \tag{5.2}$$

For a fixed l in (5.2), all constraints are linear. Then whether l is feasible for (5.2) can be determined in polynomial time. When $l_1 > 0$ and $l_2 > 0$ are both feasible for (5.2), is l feasible for any $l \in (l_1, l_2)$? If yes, we can use the binary search algorithm. Unfortunately, $\mathcal{OPT}_1(l)$ does not intersect the hyperplane $x_i = 0, i \in P$ for some l . Then the binary search algorithm is invalid here. We present the following example to illustrate it.

Example 2.

$$\begin{aligned}
& \min \quad -30x_1 - 60x_2 - 20x_3 \\
& \text{s.t.} \quad 3x_1 + 4x_2 + x_3 \leq 20, \\
& \quad \quad x_1 + 3x_2 + 2x_3 \leq 10, \\
& \quad \quad x_i \geq 0, \quad i = 1, 2, 3.
\end{aligned} \tag{5.3}$$

The optimal solution is $x^* = (4, 2, 0)^T$ and the perturbed vector for b is selected as $v = (2, 0)^T$ here. Let $P = \{3\}$.

- For fixed $l = 3$, $(0.4, 3.2, 0)^T$ which is feasible for (5.2) is an optimal solution to (2.2).
- For fixed $l = 6$, (5.2) is not feasible and there exists no optimal solution to (2.2) with the third entry being 0.

- For $l = 10$, $(0, 0, 0)^T$ is feasible for (5.2) and is an optimal solution to (2.2). The feasible solution set of (2.2) is just a singleton set including $(0, 0, 0)^T$.

Thus $l = 3$ and $l = 10$ are both feasible for (5.1) while $l = 6$ is not. That is why we consider model (1.7) instead of (5.1) in our study.

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 provided analyses of three cases in two scenarios. All models in our study are polynomial computable except for the ones containing more than one row perturbation in the constraint matrix. For those, we have developed a relatively easier binary search algorithm to get the maximum perturbation radii and whether there exist polynomial models with regard to them is still an open problem. Furthermore, this concept can also be used for the convex quadratically constrained quadratical programming(CQCQP) problem, which is considered in another paper. Also, the robust sensitivity analysis for the semidefinite programming(SDP) problem is a potential research direction.

Acknowledgments This research has been supported by National Natural Science Foundation of China Grant #11571029 and #11771243.

Reference

- [1] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, 1999.
- [2] Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on modern convex optimization*. Society for Industrial and Applied Mathematics, 2001.
- [3] Arjan B. Berkelaar, Kees Roos, and Tamás Terlaky. *The Optimal Set and Optimal Partition Approach to Linear and Quadratic Programming*. Springer US, 1997.
- [4] Dimitris Bertsimas and David B. Brown. Constructing uncertainty sets for robust linear optimization. *Operations Research*, 57(6):1483–1495, 2009.

- [5] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.
- [6] Fiona Chandra, Dennice F. Gayme, Lijun Chen, and John C. Doyle. Robustness, optimization, and architectures. *European Journal of Control*, 17(5-6):472–482, 2011.
- [7] George B. Dantzig. *Linear Programming Under Uncertainty*. INFORMS, 1955.
- [8] M. A. H Dempster. *Introduction to stochastic programming*. Springer New York, 2011.
- [9] Geir E Dullerud and Fernando G Paganini. *A Course in Robust Control Theory a convex approach*. Springer, New York, United States, 2000.
- [10] Ching Lai Hwang and Abu Syed Masud. *Multiple objective decision making-methods and applications: a state-of-the-art survey*. Springer-Verlag Berlin Heidelberg, 1979.
- [11] G. Infanger. *Planning Under Uncertainty: Solving Large-Scale Stochastic Linear Programs*. Linear Programming, 1992.
- [12] A. L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157, 1973.
- [13] Tamás Terlaky and Zhouhong Wang. On the identification of the optimal partition of second order cone optimization problems. *Siam Journal on Optimization*, 24(1):385–414, 2014.
- [14] Mi Ching Tsai and Da Wei Gu. Robust and optimal control. *Automatica*, 33(11):2095–2095(1), 2014.
- [15] Guanglin Xu and Samuel Burer. Robust sensitivity analysis of the optimal value of linear programming. *Optimization Methods and Software*, 2016. online.