Implementation of maximin and maximal solutions for linear programming problems under uncertainty

Nathan Huntley, Rolando Quiñones, Keivan Shariatmadar, Erik Quaeghebeur, Gert de Cooman

SYSTeMS Research Group, Faculty of Engineering and Architecture, Ghent University

Etienne Kerre

Fuzziness and Uncertainty Modeling, Faculty of Sciences, Ghent University

We present a software implementation of the methods for solving linear programming problems under uncertainty from previous work. Uncertainties about constraint parameters can be expressed as intervals or trapezoidal possibility distributions. The software computes the solutions for the optimality criteria maximin and maximality. For maximality with possibility distributions, only an approximate solution is obtained.

Keywords: linear programming, uncertainty, maximin, maximality, implementation

1. Introduction

Linear programming has been widely developed and used in optimization problems [1]. We consider linear programming problems with uncertainty of the form

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & Ax \leq B,\, x \geq 0, \end{array}$$

where x in $\mathbb{R}^n$ is an optimization vector, c in $\mathbb{R}^n$ is a vector of objective function coefficients, A is a $m \times n$ matrix of uncertain constraint coefficients A_{ij}, and B is an m-vector of uncertain constraint variables B_i.

In earlier work [2] we reformulated this problem into a decision problem. Uncertainty about A_{ij} and B_i is modeled by crisp intervals $[\underline{a}_{ij}, \overline{a}_{ij}]$ and $[\underline{b}_i, \overline{b}_i]$, or by possibility distributions [3]. Independence is assumed between all constraint parameters. These models can be seen as special cases of coherent lower and upper previsions $\underline{P}$, and $\overline{P}$ [4], the framework of which we use to analyze the problem. Solution methods for the *maximin* and *maximality* optimality criteria [5] were given elsewhere [6].

In this paper we discuss a Matlab implementation of these solution methods. It allows specification of the uncertainties in the A_{ij} and B_i as single values (no un-

certainty), intervals, triangular possibility distributions, and trapezoidal possibility distributions. These specifications are made in a text file, which also includes the values of m, n, and the components of c. This file is loaded into a graphical user interface, which also allows editing of the parameters and displays graphs of the possibility distributions chosen. The user then selects the type of uncertainty model to use: intervals or possibility distributions. Finally, the user chooses whether to use maximin or maximality. The interface then displays a plot of the corresponding solution and exports the solution to a file for further investigation. In Section 3 we explain the implementation and interface, and in Section 4 we give an illustration. A version of the software is available upon request.

2. Theoretical Solution

The following decision problem was introduced in earlier work [2, 6]. The gain $G_x(a,b)$ for a particular optimization vector x and a particular realization (a,b) of the constraint parameters (A,B) is c^Tx if $ax \leq b$ and L if $ax \not\leq b$. L is a penalty that must be smaller than c^Tx for any x in the *outer feasibility space*: those x for which there is a realization (a,b) of (A,B) such that $ax \leq b$.

We consider two optimality criteria for upper and lower previsions. *Maximin* selects all x that maximize the lower prevision $\underline{P}(G_x)$. *Maximality* selects all x such that, for all other potential optimization vectors z in the outer feasibility space, $\overline{P}(G_x - G_z) \geq 0$ holds. Maximality is often strengthened by requiring x to be *undominated* by any other z: either $G_x(a,b) > G_z(a,b)$ for some (a,b), or $G_x = G_z$. We cannot yet deal with this in our software, so we do not consider dominance in this paper.

The solutions in this section were derived in an earlier paper [6], which is based on a more general treatment [2].

Interval model. In the interval case the maximin problem (on the left) becomes a standard linear programming problem (on the right):

$$\begin{array}{rl} \text{maximize} & c^Tx \\ \text{subject to} & Ax \leq B,\ x \geq 0 \\ \text{with} & \underline{a} \leq A \leq \overline{a},\ \underline{b} \leq B \leq \overline{b} \end{array} \quad \rightarrow \quad \begin{array}{rl} \text{maximize} & c^Tx \\ \text{subject to} & \overline{a}x \leq \underline{b},\ x \geq 0, \end{array}$$

where $\overline{a}$ is the matrix of the $\overline{a}_{ij}$ and $\underline{b}$ is the vector of the $\underline{b}_i$.

The maximality problem becomes a vertex enumeration problem

$$\begin{array}{rl} \text{maximize} & c^Tx \\ \text{subject to} & Ax \leq B,\ x \geq 0 \\ \text{with} & \underline{a} \leq A \leq \overline{a}, \\ & \underline{b} \leq B \leq \overline{b} \end{array} \quad \rightarrow \quad \begin{array}{rl} \text{find all} & x \\ \text{subject to} & \underline{a}x \leq \overline{b},\ x \geq 0, \\ & c^Tx \geq \max_{\overline{a}z \leq \underline{b}} c^Tz. \end{array}$$

Possibility distribution model. Let the A_{ij} or B_i be modeled by trapezoidal possibility distributions π_{ij} or π_i. Special cases of trapezoidal functions are admitted: triangular functions, crisp intervals, and crisp numbers. Independence implies $\pi_A(a) := \min_{1\le i\le m, 1\le j\le n} \pi_{ij}(a_{ij})$, $\pi_B(b) := \min_{1\le i\le m} \pi_i(b_i)$, and $\pi(a,b) := \min\{\pi_A(a), \pi_B(b)\}$. The maximin problem becomes a nested optimization problem

$$\begin{array}{rl} \text{maximize} & c^T x \\ \text{subject to} & Ax \le B, \\ & x \ge 0 \\ \text{with} & \text{given } \pi \end{array} \quad \rightarrow \quad \begin{array}{ll} \text{maximize} & L + (1-t) \left(\begin{array}{ll} \text{maximize} & c^T x - L \\ \text{subject to} & \overline{a}^t x \le \underline{b}^t, \\ & x \ge 0 \end{array} \right), \\ \text{subject to} & 0 \le t < 1 \end{array}$$

where $\overline{a}^t_{ij} = \max\{a_{ij} \colon \pi_{ij}(a_{ij}) \ge t\}$, $\underline{b}^t_i = \min\{b_i \colon \pi_i(b_i) \ge t\}$, $\overline{a}^t$ is the matrix with components $\overline{a}^t_{ij}$, and $\underline{b}^t$ is the vector with components $\underline{b}^t_i$.

For maximality, $\overline{P}(G_x - G_z)$ is difficult to calculate, and is of the form

$$\overline{P}(G_x - G_z) = \alpha(x,z,\pi) + \beta(x,z)\gamma(x,z,\pi), \tag{1}$$

where α is an easily-calculated function, β is an easily-calculated positive function, and γ is a function taking values in $[0,1]$ that we have found no efficient way to calculate. In the next section we explain how the software deals with this problem.

3. Software Implementation

Methods. For the interval case, we solve the linear programming and vertex enumeration problems using standard toolboxes. For maximin with possibility distributions, the inner maximization is a linear program. The outer maximization is unimodal in t, so we use the golden section method [7, § 10.2]

For maximality, the upper prevision in Eq. (1) is difficult to calculate efficiently. However, we only need to know whether it is nonnegative, that is, whether

$$\gamma(x,z,\pi) \ge -\alpha(x,z,\pi)/\beta(x,z). \tag{2}$$

Since $\gamma(x,z,\pi) \in [0,1]$, if the right-hand side is 0 or less, then the left-hand side is not smaller. If the right-hand side is greater than 1, then the left hand side is smaller. If the right-hand side is in $(0,1]$, it turns out we can check Eq. (2) by solving at most m linear programs.

To find an approximate set of maximal points, we discretize the outer feasible region into a grid of candidate vectors, and test these against one another. As soon as a candidate is found to be non-maximal it can be removed from all further comparisons, because the ordering corresponding to maximality is transitive. Any candidate removed by this method is non-optimal, but a candidate x not removed is not necessarily maximal: it could be that none of the z such that $\overline{P}(G_x - G_z) < 0$ were in the grid. This can often happen when the grid resolution is too coarse or when the candidate vector is close to a maximal vector.

Interface and Input. On the right, we show an example input file, used for the illustration of Section 4. `n` sets the dimension of x. `m` sets the number of constraint rows. The numbers under `Objective function` give the values of the c_i. Each row of numbers under `A` gives the uncertainty model for a particular element A_{ij}: in this case (in order) A_{11}, A_{12}, A_{13}, all of which here are triangular possibility distributions. Each row of numbers under `B` gives the uncertainty model for a particular B_i, in this case B_1 only, which here is a crisp value 8. As an example, adding another constraint row would mean there are four new uncertain values: A_{21}, A_{22}, A_{23}, and B_2. To add these to the problem, we would add three more rows of numbers underneath `8.3 9.3 10.4` (corresponding to A_{21}, A_{22}, A_{23}), and one row of numbers underneath `8` (corresponding to B_2). `L` is the penalty L, and `resolution` is the grid resolution for the method for possibility distributions and maximality. These last two are optional.

```
Input File
m=1
n=3
Objective function
-5 -3.75 -2.5
A
4.8 5 5.3
8 9 10
8.3 9.3 10.4
B
8
resolution=40
L=-10
```

In Figures 1–3 we show the three panels of the graphical user interface.

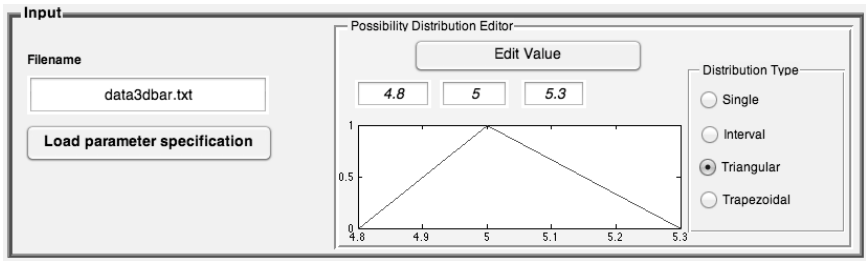


Figure 1. First GUI panel: the input file is loaded, and the distributions are viewed and edited.

Figure 2. Second GUI panel: the values for c and the parameters for the possibility distributions of A and B are all shown. Clicking on a cell displays the possibility distribution in the possibility distribution editor (see Fig. 1).

4. Illustration: minimizing the cost of a beam

We apply the method to a loaded beam problem. We want to minimize the total cost of a beam of length ℓ and cross section area a that may consist of n different segments i of different material, each with a relative cost per meter of c_i. Under a load f, the beam should not stretch more than a prescribed length d. So we must

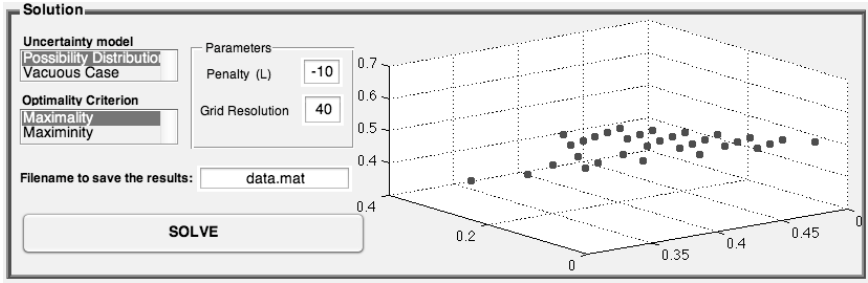
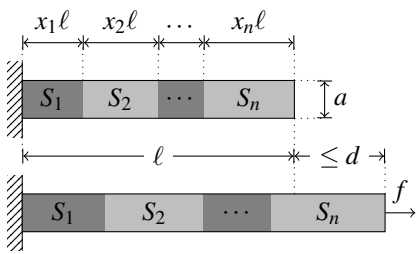


Figure 3. Third GUI panel: the choice of uncertainty model to use can be made (if 'Vacuous Case' is chosen, then all triangular and trapezoidal possibility distributions are reduced to crisp intervals). When 'Possibility Distribution' is chosen, the optimality criterion to use must be selected. When 'Vacuous Case' is chosen, solutions for both criteria are calculated. When n is 2 or 3, a plot of the solution is displayed in the right panel. Finally, the solution is exported to a specified file, along with c, L, and the distributions for the A_{ij} and B_i.

find relative segment lengths x_i that minimize the cost. The constraint can be approximated by a linear one determined by a finite element analysis [8] in which we take beam segments as elements. There is uncertainty about the elastic compliances S_i (inverses of elastic or Young moduli) that determine how much each material stretches. The problem can be formulated as the linear program with one uncertain constraint on the right.



$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} -c_i \ell x_i \\ \text{subject to} & \sum_{i=1}^{n} S_i x_i \leq \frac{da}{f\ell}, \\ & \sum_{i=1}^{n} x_i = 1, x \geq 0. \end{array}$$

We imagine a beam with $n = 3$ segments with the following materials, using triangular possibility distributions for the elastic compliances (units are mm^2/MN): wrought iron with $S_1 \sim (4.8; 5.0; 5.3)$; brass with $S_2 \sim (8.0; 9.0; 10.0)$; bronze with $S_3 \sim (8.3; 9.3; 10.4)$. The quantities ℓ, a, f, and d are chosen such that $\frac{da}{f\ell} = 8.0\text{mm}^2/\text{MN}$. The relative costs are $3c_1 = 4c_2 = 6c_3$ and the penalty is $L = -10$. The results of the analysis of this problem with our program are given in Fig. 4.

5. Conclusion

We implemented the approaches in previous work [6] for linear programming problems under uncertainty. An intuitive user interface allows visualization and specification of the uncertain parameters, and offers a choice of two solution methods (maximin and maximality) for two uncertainty models (intervals and possibility distributions). The current version allows uncertainties to be given as any combina-

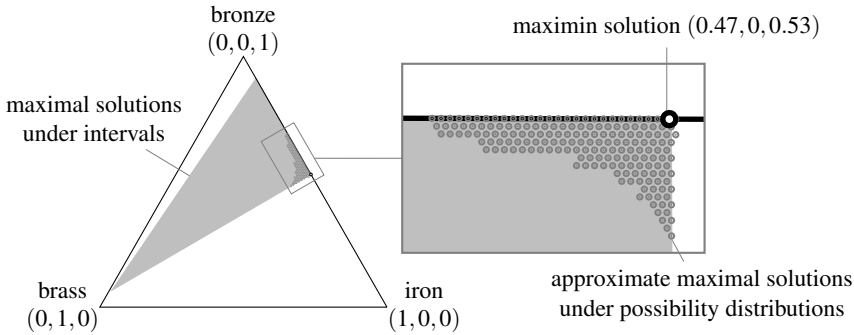


Figure 4. Solution of beam problem for interval and possibility distribution cases. Each point in the simplex corresponds to the optimization vector x with that ratio of materials. The light grey region is the maximal region using intervals. The dark grey points show the approximate maximal region using possibility distributions. The maximin solution is the same for both uncertainty models.

tion of crisp values, intervals, and triangular or trapezoidal possibility distributions. Future goals are to include other uncertainty models, to address dominance, and to find an efficient method for maximality with possibility distributions for large n.

Acknowledgements

Supported by the IWT SBO project 60043, “Fuzzy Finite Element Method”. We thank the reviewers for their helpful comments.

References

1. G. Dantzig, *Linear Programming and Extensions* (1998).
2. E. Quaeghebeur, K. Shariatmadar and G. De Cooman, *Fuzzy Sets and Systems* (in press).
3. D. Dubois and H. Prade, *Possibility Theory* (1988).
4. P. Walley, *Statistical Reasoning with Imprecise Probabilities* (1991).
5. M. C. M. Troffaes, *International Journal of Approximate Reasoning* **45**, 17 (2007).
6. E. Quaeghebeur, N. Huntley, K. Shariatmadar and G. De Cooman, Maximin and maximal solutions for linear programming problems with possibilistic uncertainty, accepted by IPMU-2012.
7. W. Press, S. Teukolsky, W. Vetterling and B. Flannery, *Numerical Recipes* (2007).
8. O. Zienkiewicz, R. L. Taylor and J. Z. Zhu (eds.), *The Finite Element Method: Its Basis and Fundamentals* (2005).