

Linear Programming with variable matrix entries

by

*Paolo Serafini, University of Udine, Italy
Department of Mathematics and Computer Science*

Abstract: We consider linear programming (continuous or integer) where some matrix entries are decision parameters. If the variables are nonnegative the problem can be easily solved in two phases. It is shown that direct costs on the matrix entries make the problem NP-hard. Finally a strong duality result is provided.

Keywords: linear programming, integer linear programming, variable data.

1. INTRODUCTION

There are linear programming problems where some constraint data are not precisely fixed but can vary within some bounds and may be considered design parameters. This is different from the situation where the input data are not known precisely and they are only assumed to belong to some tolerance interval. This type of sensitivity analysis has been subject of considerable analysis and reference can be found for instance in [3], [4], [5], [6] and [8].

In this paper we deal with the case when some of the data can be fixed by the decision maker to a precise value within a prescribed interval. In this sense we speak of design parameters rather than uncertain data. The diet problem (considered also in [4] in relation to interval data) is an instance of problems of this type: there are nutrients (y), ingredients (x) and dishes (z). Ingredients and nutrients are linked together through a fixed data matrix A as $Ax = y$. Ingredients and dishes are linked together through a data matrix R consisting of recipes as $Rz = x$. The variables z related to dishes are typically integer, but recipe data in R can usually vary within some bounds without affecting the particular taste of a dish. Therefore we would like to consider as decision variables not

Corresponding author: Paolo Serafini, email: serafini@dimi.uniud.it, postal address: Dipartimento di Matematica e Informatica, Via delle Scienze 206, 33100 Udine, Italy

only y , x and z but also the matrix R . For a practical model it is important to be able to take care of this variability.

General blending problems present such data variability. If the variability is independent in each data entry, the variables are nonnegative and there is no direct cost on the matrix entries, then the nonlinear model, which results by varying simultaneously both the actual variables and the matrix entries, can be solved via linear programming in two phases. If the objective function includes the variable matrix entries the problem becomes NP-hard. To the best of our knowledge this analysis, although simple, has not been investigated in the literature.

The paper is organized as follows: in Section 2 the main result is stated, in Section 3 a particular way of computing the matrix entries is proposed, in Section 4 a small example is shown to illustrate the procedure, in Section 5 it is shown that introducing direct costs on the matrix entries makes the problem difficult and finally in Section 6 a duality result is presented.

2. PROBLEM STATEMENT

Let us consider the following problem

$$\begin{aligned} \min \quad & c x \\ & A x = b \\ & x \in P \subset \mathbb{R}^n \end{aligned} \tag{1}$$

where c is an n row vector, b is an m column vector, x is an n column vector, A is an $m \times n$ matrix and P is any subset of \mathbb{R}^n (including finite sets). The entries of the matrix A are not fixed, but they can assume any value within prescribed intervals, i.e.

$$a_{ij}^- \leq a_{ij} \leq a_{ij}^+ \tag{2}$$

In other words the feasible set for x is

$$X := \bigcup_{A^- \leq A \leq A^+} \{x \in P : A x = b\}$$

(where $A^- \leq A \leq A^+$ is meant entry-wise like in (2)) and we want to solve $\min_{x \in X} c x$. In general X is not convex and therefore the minimization of a linear functional on X can be very hard even if P does not contain hard constraints. However, if we assume that P is contained in the positive orthant (in fact any orthant works well), X is convex and the problem can be solved in two phases: first compute the optimum \hat{x} of

$$\begin{aligned} \min \quad & c x \\ & A^- x \leq b \\ & A^+ x \geq b \\ & x \in P \end{aligned} \tag{3}$$

and then find a matrix A such that

$$A^- \leq A \leq A^+, \quad A \hat{x} = b \quad (4)$$

Let \bar{X} be the feasible set of (3). We are going to show that, in the assumption of nonnegativity of x , X and \bar{X} coincide.

Theorem 1: $X = \bar{X}$ if $P \subset \mathbb{R}_+^n$.

Proof: If $x \in X$, by definition there exists a matrix A such that $Ax = b$ and $A^- \leq A \leq A^+$. Hence from $A - A^- \geq 0$ and $x \geq 0$ one has $(A - A^-)x \geq 0$, i.e. $b \geq A^-x$. For the other constraint the reasoning is similar and so we have $x \in \bar{X}$.

If $x \in \bar{X}$ we have to show the existence of a matrix A feasible in (4). Let A_i^- and A_i^+ be the i -th rows of the respective matrices. Let us consider the segment in \mathbb{R}^n joining the points A_i^- and A_i^+ . By continuity there exists a point on the segment where the functional x (the functional is defined as $x : \mathbb{R}^n \rightarrow \mathbb{R}, x : y \mapsto \sum_{j=1}^n y_j x_j$) assumes the value b_i (since $A_i^-x \leq b_i$ and $A_i^+x \geq b_i$). Let A_i be that particular value. Since A_i is on the segment we have $A_i^- \leq A_i \leq A_i^+$. Then the matrix A is obtained by patching the rows A_i . ■

Note that the proof can be slightly modified to show that the result holds on any particular orthant. Hence, if P is convex, X restricted to a generic orthant is convex.

Without the assumption of nonnegativity X is not convex in general. In the simple example $ax_1 + x_2 = 0$, with $a^- \leq a \leq a^+$, X is a cone joined to its opposite cone. Indeed the problem is difficult in general as shown by the following result:

Theorem 2: *Problem (1) is NP-hard in general.*

Proof: Just note that

$$\bigcup_{-1 \leq a \leq 1} \{x : ax = 1, -1 \leq x \leq 1\} = \bigcup_{a \in \{-1, 1\}} \{x : ax = 1, -1 \leq x \leq 1\} = \{-1, 1\}$$

Therefore any 0-1 linear programming problem can be first easily transformed into a (-1,1) linear programming problem and then each constraint $x_j \in \{-1, 1\}$ can be transformed into $a_j x_j = 1, -1 \leq x_j \leq 1$, with $a_j \in [-1, 1]$. ■

As can be seen from the proof of Theorem 1 the assumption of independent variability in the data entry is essential to show the existence of a feasible matrix A . Hence the standard trick of converting unrestricted variables into nonnegative ones does not work here because this transformation makes some matrix entries dependent on others.

In some problems particular entries may be related. For instance they may refer to the same physical quantity and hence they must assume the same value. If entries of this kind are on different rows there is no guarantee that such a feasible matrix exists and the approach shown here does not work. However, there are problems of this type which can be easily modeled in alternative ways. For instance suppose that the entries of a particular column must retain their mutual ratios (this can be practically relevant), so that a column

A^j is feasible if $A^j = d \bar{A}^j$ and $d^- \leq d \leq d^+$, with d scalar and \bar{A}^j assigned. Then we may substitute $A^j x_j$ with $\bar{A}^j y_j$ and add the constraints $d^- x_j \leq y \leq d^+ x_j$.

3. COMPUTING THE MATRIX ENTRIES

The theorem allows to solve $\min_{x \in X} c x$ by solving (3). However, solving (3) does not provide a matrix A satisfying (4). Since there can be many feasible solutions to (4) we may consider finding a matrix A satisfying certain properties. One simple requirement could be that the entries are as much close as possible to a prescribed value \bar{a}_{ij} (for instance $\bar{a}_{ij} := (a_{ij}^+ + a_{ij}^-)/2$). Toward this goal several alternative methods are available.

First let us observe that in any case the problem can be decomposed into independent problems, one for each row i . Moreover, we do not have to compute the values which are fixed (i.e. $a_{ij}^- = a_{ij} = a_{ij}^+ =: \bar{a}_{ij}$) and those which are irrelevant because they multiply a zero value of \hat{x}_j . For notational simplicity we write the formulas as if all values need to be computed.

One possible approach is based on the minimization of

$$\min \sum_j |a_{ij} - \bar{a}_{ij}|$$

so that we have the following linear programming problems (one for each i).

$$\begin{aligned} \min \quad & \sum_j w_{ij}^+ + w_{ij}^- \\ & w_{ij}^+ \geq a_{ij} - \bar{a}_{ij} \quad \forall j \\ & w_{ij}^- \geq \bar{a}_{ij} - a_{ij} \quad \forall j \\ & \sum_j a_{ij} \hat{x}_j = b_i \\ & a_{ij}^- \leq a_{ij} \leq a_{ij}^+ \quad \forall j \\ & w_{ij}^+ \geq 0, w_{ij}^- \geq 0 \quad \forall j \end{aligned}$$

Perhaps better objective functions (with matrix entries less “attracted” by the interval extremes) are quadratic ones, like:

$$\min \sum_j (a_{ij} - \bar{a}_{ij})^2$$

These are continuous quadratic knapsack problems first investigated by Brucker [2] and later independently by [1] and [7]. In [2] it is shown that the problem can be solved in time $O(n)$. Hence the second phase can be solved in linear time with respect to the number of matrix entries.

4. AN EXAMPLE

For the sake of illustration of the procedure let us consider the following multiknapsack problem

$$\begin{aligned}
\max \quad & \sum_{j=1}^n v_j z_j + \sum_{i=1}^m u_i x_i \\
& \sum_{j=1}^n w_{ij} z_j = x_i \quad i = 1, \dots, m \\
& 0 \leq x_i \leq c_i \quad i = 1, \dots, m \\
& z_j \in \{0, 1\} \quad j = 1, \dots, n \\
& w_{ij}^- \leq w_{ij} \leq w_{ij}^+ \quad i = 1, \dots, m, j = 1, \dots, n
\end{aligned}$$

with data $n = 5$, $m = 2$, $v = (10, 8, 6, 9, 5)$, $u = (4, 3)$, $c = (20, 30)$,

$$w^- = \begin{pmatrix} 6 & 5 & 8 & 5 & 3 \\ 5 & 9 & 12 & 8 & 10 \end{pmatrix}, \bar{w} = \begin{pmatrix} 7 & 8 & 8 & 6 & 3 \\ 7 & 11 & 13 & 8 & 10 \end{pmatrix}, w^+ = \begin{pmatrix} 8 & 9 & 9 & 7 & 4 \\ 8 & 12 & 15 & 8 & 10 \end{pmatrix}$$

with \bar{w} the preferred values for the matrix entries. The first phase consists in solving

$$\begin{aligned}
\max \quad & \sum_{j=1}^n v_j z_j + \sum_{i=1}^m u_i x_i \\
& \sum_{j=1}^n w_{ij}^- z_j \leq x_i \quad i = 1, \dots, m \\
& \sum_{j=1}^n w_{ij}^+ z_j \geq x_i \quad i = 1, \dots, m \\
& 0 \leq x_i \leq c_i \quad i = 1, \dots, m \\
& z_j \in \{0, 1\} \quad j = 1, \dots, n
\end{aligned}$$

The solution is $\hat{z} = (1, 0, 1, 1, 0)$, $\hat{x} = (20, 30)$. In order to compute \hat{w} (only for the index set $\{(1, 1), (1, 3), (1, 4), (2, 1), (2, 3)\}$) we use the quadratic model, apply the algorithm by Brucker and get

$$\hat{w} = \begin{pmatrix} 6.5 & 8 & 8 & 5.5 & 3 \\ 8 & 11 & 14 & 8 & 10 \end{pmatrix}$$

5. COST COEFFICIENTS ON THE MATRIX ENTRIES

The model we have described does not consider direct costs to the variable matrix entries. This does not mean that changing the value of a matrix entry is costless. In the

diet example with nutrients y , ingredients x , dishes z , linked together as $Ax = y$ and $Rz = x$, the variable data are the recipe data in R . Changing R affects indirectly the cost because x is changed and x does enter the objective function.

If we want to assess anyway a cost on the possibility of changing the matrix entries with respect to some preferred matrix \hat{A} , then we might measure the effect of the change on x by considering $|b - \hat{A}x|$.

However, if we do need direct cost coefficients on the matrix entries, this request makes the problem NP-hard. To show this fact consider the following recognition version of the 0-1 linear programming problem: is there an $x \in \{0, 1\}^n$ such that $Ax \leq b$? We want to transform this problem into the following: given a number K , a matrix A , with some entries specified only by their lower and upper bounds, i.e. $a_{ij}^- \leq a_{ij} \leq a_{ij}^+$ for (i, j) in a specified set M , cost coefficients c_j , $j := 1, \dots, n$, d_{ij} , $(i, j) \in M$, are there $x_j \geq 0$, $a_{ij}^- \leq a_{ij} \leq a_{ij}^+$, $(i, j) \in M$, such that $\sum_j c_j x_j + \sum_{(i,j) \in M} d_{ij} a_{ij} \geq K$ and $Ax = b$?

The transformation goes as follows. First we consider an intermediate transformation through $x = 2y - 1$. Then feasibility of $x \in \{0, 1\}^n$, $Ax \leq b$, is equivalent to feasibility of

$$\begin{aligned} 2Ay &\leq b + A\mathbf{1} \\ y &\in \{1/2, 1\} \end{aligned} \tag{5}$$

where $\mathbf{1}$ is the all-one vector. Now let us consider the following particular instances of the problem under investigation:

$$\begin{aligned} \max \quad & \sum_j c_j + y_j \\ & 2c_j y_j = 1 \quad \forall j \\ & 2Ay \leq b + A\mathbf{1} \\ & 1/2 \leq y_j \leq 1 \quad \forall j \\ & 1/2 \leq c_j \leq 1 \quad \forall j \end{aligned} \tag{6}$$

asking the question whether there are solutions in (6) with objective function value at least $K = 3/2n$.

Let us suppose that there is such a solution in (6). Note that for each j the sum $c_j + y_j$ is upper bounded by $3/2$ and this value is reached either with $c_j = 1/2$, $y_j = 1$ or $c_j = 1$, $y_j = 1/2$. For all other admissible values of y_j and c_j one has $c_j + y_j < 3/2$. This means that each pair (y_j, c_j) in the sum $\sum_j c_j + y_j$ must sum up to $3/2$. Therefore if there is a solution with objective value at least $3/2n$ (in fact equal to $3/2n$) this must be such that $y_j \in \{1/2, 1\}$ implying that there is a feasible solution to $x \in \{0, 1\}^n$, $Ax \leq b$.

Suppose now that there is a feasible solution to $x \in \{0, 1\}^n$, $Ax \leq b$. This clearly provides a solution to (6) with value $3/2n$.

6. DUALITY ANALYSIS

Let us consider the following dual pair of problems

$$\begin{aligned} v(A) = \min \quad & c x & d(A) = \max \quad & y b \\ & A x = b & & y A \leq c \\ & x \geq 0 & & \end{aligned} \quad (7)$$

If $\mathbf{A} = \{A : A^- \leq A \leq A^+\}$ is the set of matrices we are allowed to use in (7) then problem (1) (with respect to (7)) is

$$\hat{v} := \min_{A \in \mathbf{A}} v(A) = \min_{A \in \mathbf{A}} d(A)$$

Note that \hat{v} may be equivalently expressed as

$$\hat{v} = \min \quad c x \\ x \in \bigcup_{A \in \mathbf{A}} \{x : A x = b, x \geq 0\}$$

So we have:

$$\min_{A \in \mathbf{A}} d(A) = \min_{A \in \mathbf{A}} \max_{y : y A \leq c} y b = \min_{A \in \mathbf{A}} \max_y y b - \delta(c - y A) \geq \max_y \min_{A \in \mathbf{A}} y b - \delta(c - y A)$$

with $\delta : \mathbb{R}^n \rightarrow \mathbb{R}$ defined as

$$\delta(x) := \begin{cases} 0 & \text{if } x_j \geq 0, \forall j \\ +\infty & \text{otherwise} \end{cases}$$

The minimax theorem cannot be applied due to the lack of convexity of the function $A \mapsto y b - \delta(c - y A)$ (the stated inequality always holds). Since

$$\min_{A \in \mathbf{A}} y b - \delta(c - y A) = \begin{cases} y b & \text{if } y A \leq c \text{ for all } A \in \mathbf{A} \\ -\infty & \text{otherwise} \end{cases}$$

we have

$$\min_{A \in \mathbf{A}} d(A) \geq \max \{y b : y A \leq c, \forall A \in \mathbf{A}\} =: \hat{d}$$

Note that \hat{d} may be equivalently expressed as

$$\begin{aligned} \hat{d} = \max \quad & y b \\ & y \in \bigcap_{A \in \mathbf{A}} \{y : y A \leq c\} \end{aligned} \quad (8)$$

We can actually prove that $\hat{v} = \hat{d}$, thus providing a strong duality result between the following pair of problems

$$\begin{aligned} \hat{v} = \min \quad & c x & \hat{d} = \max \quad & y b \\ & x \in \bigcup_{A \in \mathbf{A}} \{x : A x = b, x \geq 0\} & & y \in \bigcap_{A \in \mathbf{A}} \{y : y A \leq c\} \end{aligned}$$

Theorem 3: $\hat{v} = \hat{d}$.

Proof: We have already proved that

$$\begin{aligned} \hat{v} = \min \quad & c x \\ & A^- x \leq b \\ & A^+ x \geq b \\ & x \geq 0 \end{aligned}$$

whose dual is

$$\begin{aligned} \hat{v} = \hat{d} := \max \quad & (y^+ - y^-) b \\ & y^+ A^+ - y^- A^- \leq c \\ & y^+ \geq 0, y^- \geq 0 \end{aligned} \quad (9)$$

It is enough to show that the projection onto the subspace $y = y^+ - y^-$ of the feasible set of (9) coincides with the feasible set of (8). Let y^+ and y^- be feasible in (9). Then for any $A \in \mathbf{A}$ we have

$$y A = (y^+ - y^-) A = y^+ A - y^- A \leq y^+ A^+ - y^- A^- \leq c$$

Conversely let y be feasible in (8). Let $I^+(y)$ and $I^-(y)$ be the matrices derived from the identity matrix by defining its diagonal elements as

$$I_i^+(y) := \begin{cases} 1 & \text{if } y_i \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad I_i^-(y) := \begin{cases} 1 & \text{if } y_i < 0 \\ 0 & \text{otherwise} \end{cases}$$

Then define

$$y^+ := y I^+(y) \quad y^- := -y I^-(y)$$

thus having

$$y^+ A^+ - y^- A^- = y (I^+(y) A^+ + I^-(y) A^-) \leq c$$

where the last inequality follows from the fact that $I^+(y) A^+ + I^-(y) A^- \in \mathbf{A}$. ■

In view of this result the optimal dual variables \hat{y}^+ and \hat{y}^- computed by solving (3) (with P the nonnegative orthant) provide a variable $\hat{y} := \hat{y}^+ - \hat{y}^-$ which is optimal for the problem (8) and dual optimal for the problems

$$\begin{aligned} \min \quad & c x & \max \quad & y b \\ & \hat{A} x = b & & y \hat{A} \leq c \\ & x \geq 0 & & \end{aligned} \quad (10)$$

with \hat{A} the final matrix obtained in the second phase of the procedure.

7. REFERENCES

[1] VAN DEN BOSCH, P.P.J. AND F.A. LOOTSMA, "Scheduling of power generation via large-scale nonlinear optimization", *Journal of Optimization Theory and Applications*, **55**, 1987, 313-326.

- [2] BRUCKER, P., “An $O(n)$ algorithm for quadratic knapsack problems”, *Operations Research Letters*, **3**, 1984, 163-166.
- [3] HANSEN, E., *Global optimization using interval analysis*, Marcel Dekker, New York, 1992.
- [4] JANSSEN, C., “A self-validating method for solving linear programming problems with interval input data”, *Comput. Suppl.*, **6**, 1988, 33-46.
- [5] JANSSEN, C., “On self-validating methods for optimization problems”, *Stud. Comput. Math.*, **5**, 1993, 381-438.
- [6] LODWICK, W. A. AND K. D. JAMISON, “Interval methods and fuzzy optimization”, *Internat. J. Uncertain. Fuzziness Knowledge-Based Systems*, **5**, 1997, 239-249.
- [7] PARDALOS, P.M. AND N. KOVOOR, “An algorithm for a single constrained class of quadratic programs subject to upper and lower bounds”, *Mathematical Programming*, **46**, 1990, 321-328.
- [8] ROKNE, J. G., “Interval arithmetic and interval analysis: an introduction”, *Stud. Fuzziness Soft. Comput.*, **70**, 2001, 1-22.