Modeling and Solving Linear Programming with

Jose M. Sallan, Oriol Lordan, Vicenc Fernandez

open access

OmniaScience

Scholar

Modeling and solving linear programming with R

Jose M Sallan Oriol Lordan Vicenc Fernandez

Modeling and Solving Linear Programming with R

Authors:

Jose M. Sallan, Oriol Lordan, Vicenc Fernandez

Universitat Politècnica de Catalunya

ISBN: 978-84-944229-3-5

DOI: http://dx.doi.org/10.3926/oss.20

Cover image: Tobias Wolf

Contents

CHAPTER 1

Introduction

This book is about using linear programming to help making better decisions in the organizational context. Linear programming is one of the most useful and extensively used techniques of operational research. It is one special case of mathematical optimization, where the function to optimize and the constraints are linear functions of the decision variables. Posterior developments of linear programming include the possibility of defining some decision variables as integer, widening the range of problems solvable by linear programming considerably.

This is the first of a series of books that act as a support of a pedagogical program based on teaching operational research techniques with R. R [6] is a programming language and software environment for statistical computing and graphics. The R language is widely used among statisticians and data miners for developing statistical software and data analysis. It is an open source programming environment, that runs in most operating systems. The strength of R comes from the large num-

ber of libraries developed by a lively community of software developers. Within the context of this teaching program, the objective of this book is twofold. On the one side, our aim is to present a pragmatic introduction to linear programming, presenting through practical examples the possibilities of modeling through linear programming situations of decision making in the organizational context. On the other side, some libraries to solve linear programming models are presented, such as Rglpk [7], lpSolve [1] and Rsymphony [3].

To achieve these aims, the book is organized as follows. In 2.6.2 are developed the basics of linear programming: an introduction of formulation of linear models, an introduction to the features of the optimum of a linear program, including duality analysis, and to the formulation and solution of linear programs including integer variables. The chapter concludes with an introduction to the use of linear programming solvers in R.

chapter 3 includes ten optimization problems solvable by linear programming. Each of the problems is presented with the following structure: after presenting the problem, a solution through linear programming is offered. Then we show how to solve the problem in R. There are several ways to parse a problem into a R solver. In this collection of problems, we show how to use a standard linear programming syntax, such as CPLEX, and how to enter the model using the R syntax.

We have chosen to use online resources to keep this book updated. In http://bit.ly/1zkJpVw we are keeping a list of linear programming solvers, together with its implementation in R. We encourage readers to send us a comment if they find the information incomplete or not updated. All the source code used in this book is stored and updated in the

https://github.com/jmsallan/linearprogramming GitHub repository.

We hope that this book becomes a valuable resource to everybody interested in a hands-on introduction to linear programming, that helps to reduce the steep of the learning curve to implement code including resolution of linear programming models.

CHAPTER 2

Solving linear programming

2.1 An introduction to linear programming

Linear programming is one of the most extensively used techniques in the toolbox of quantitative methods of optimization. Its origins date as early as 1937, when Leonid Kantorovich published his paper *A new method of solving some classes of extremal problems*. Kantorovich developed linear programming as a technique for planning expenditures and returns in order to optimize costs to the army and increase losses to the enemy. The method was kept secret until 1947, when George B. Dantzig published the simplex method for solving linear programming [2]. In this same year, John von Neumann developed the theory of duality in the context of mathematical analysis of game theory.

One of the reasons for the popularity of linear programming is that it allows to model a large variety of situations with a simple framework.

Furthermore, a linear program is relatively easy to solve. The simplex method allows to solve most linear programs efficiently, and the Karmarkar interior-point methods allows a more efficient solving of some kinds of linear programming.

The power of linear programming was greatly enhanced when came the opportunity of solving integer and mixed integer linear programming. In these models all or some of the decision variables are integer, respectively. This field was opened by the introduction of the branch and bound method by Land and Doig. Later other algorithms have appear, like the cutting plane method. These techniques, and the extension of computing availability, have increased largely the possibilities of linear programming.

In this chapter we will provide a brief introduction to linear programming, together with some simple formulations. We will also provide an introduction to free software to solve linear programming in R, in particular the R implementations of lp_solve and GLPK through the libraries lpSolve, Rglpk and Rsymphony, among others. chapter 3 introduces some applications of linear programming, through a collection of solved linear programming problems. For each problem a posible solution through linear programming is introduced, together with the code to solve it with a computer and its numerical solution.

2.2 Linear programming formulation

2.2.1 The structure of a linear program model

Roughly speaking, the linear programming problem consists in *optimizing* (that is, either minimize or maximize) the value of a linear *objective function* of a vector of *decision variables*, considering that the variables can only take the values defined by a set of linear *constraints*. Linear programming is a case of *mathematical programming*, where objective function and constraints are linear.

A formulation of a linear program in its canonical form of maximum is:

$$
\begin{aligned}
&\text{MAX } z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
&\text{s. t. } a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 \\
&\qquad a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2 \\
&\qquad \cdots \\
&\qquad a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m \\
&\qquad x_i \geq 0
\end{aligned}
$$

The model has the following elements:

- An *objective function* of the n decision variables x_j. Decision variables are affected by the *cost coefficients* c_j
- A set of m *constraints*, in which a linear combination of the variables affected by coefficients a_{ij} has to be less or equal than its *right hand side value* b_i (constraints with signs greater or equal or equalities are also possible)

- The bounds of the decision variables. In this case, all decision variables have to be nonnegative.

The constraints of the LP define the *feasible region*, which is the set of values that satisfy all constants. For a LP of n variables, the feasible region is a n-dimensional convex polytope. For instance, for $n = 2$ the feasible region is a convex polygon.

The LP formulation shown above can be expressed in matrix form as follows (cap bold letters are matrices and cap small bold letters are column vectors):

$$\begin{aligned} \text{MAX } z &= \mathbf{c}'\mathbf{x} \\ \text{s. t. } \mathbf{A}\mathbf{x} &\leq \mathbf{b} \\ \mathbf{x} &\geq 0 \end{aligned}$$

Using the same matrix syntax, we can write the *canonical form* of minimum of a linear program as:

$$\begin{aligned} \text{MIN } z &= \mathbf{c}'\mathbf{x} \\ \text{s. t. } \mathbf{A}\mathbf{x} &\geq \mathbf{b} \\ \mathbf{x} &\geq 0 \end{aligned}$$

Another usual way to express a linear program is the *standard form*. This form is required to apply the simplex method to solve a linear program. Here we have used OPT to express that this form can be defined for maximum or minimum models.

$$\begin{aligned} \text{OPT } z &= \mathbf{c}'\mathbf{x} \\ \text{s. t. } \mathbf{A}\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &\geq 0 \end{aligned}$$

An additional condition to use the simplex method is that righthand side values $\mathbf{b} \geq 0$. All other parameters are not restricted in sign.

2.2.2 A simple example of a PL model

Let's consider the following situation:

A small business sells two products, named Product 1 and Product 2. Each tonne of Product 1 consumes 30 working hours, and each tonne of Product 2 consumes 20 working hours. The business has a maximum of 2,700 working hours for the period considered. As for machine hours, each tonne of Products 1 and 2 consumes 5 and 10 machine hours, respectively. There are 850 machine hours available.

Each tonne of Product 1 yields 20 M€ of profit, while Product 2 yields 60 M€ for each tonne sold. For technical reasons, the firm must produce a minimum of 95 tonnes in total between both products. We need to know how many tonnes of Product 1 and 2 must be produced to maximize total profit.

This situation is apt to be modeled as a PL model. First, we need to define the *decision variables*. In this case we have:

- $P1$ number of tonnes produced and sold of Product 1
- $P2$ number of tonnes produced and sold of Product 2

The *cost coefficients* of these variables are 20 and 60, respectively. Therefore, the *objective function* is defined multiplying each variable by its corresponding cost coefficient.

The constraints of this LP are:

- A constraint WH making that the total amount of working hours used in Product 1 and Product 2, which equals $30P1 + 20P2$, is less or equal than 2,700 hours.
- A similar constraint MH making that the total machine hours $5P1 + 10P2$ are less or equal than 850.
- A PM constraint making that the total units produced and sold $P1 + P2$ are greater or equal than 95.

Putting all this together, and considering that the decision variables are nonnegative, the LP that maximizes profit is:

$$\begin{aligned} \text{MAX } & z = 20P1 + 60P2 \\ \text{s.t. WH)} \ & 30P1 + 20P2 \leq 2700 \\ \text{MH} \ & 5P1 + 10P2 \leq 850 \\ \text{PM)} \ & P1 + P2 \geq 95 \\ & P1 \geq 0, \ P2 \geq 0 \end{aligned}$$

2.2.3 A transportation problem

Let's consider a *transportation problem* of two origins a and b, and three destinations 1, 2 and 3. In Table 2.1 are presented the cost c_{ij} of transporting one unit from the origin i to destination j, and the maximal capacity of the origins and the required demand in the destinations. We need to know how we must cover the demand of the destinations at a minimal cost.

	1	2	3	capacity
a	8	6	3	70
b	2	4	9	40
demand	40	35	25	

Table 2.1: Parameters of the transportation problem

This situation can be modeled with a LP with the following elements:

- Decision variables of the form x_{ij}, representing units transported from origin i to destination j
- An objective function with cost coefficients equal to c_{ij}
- Two sets of constraints: a less or equal set of constraints for each origin, limiting the units to be transported, and a greater of equal set of constraints representing that the demand of each destination must be covered.

The resulting LP is:

$$\begin{aligned}
\text{MIN } & z = 8x_{a1} + 6x_{a2} + 3x_{a3} + 2x_{b1} + 4x_{b2} + 9x_{b3} \\
\text{s.a. ca) } & x_{a1} + x_{a2} + x_{a3} \leq 70 \\
\text{cb) } & x_{b1} + x_{b2} + x_{b3} \leq 40 \\
\text{d1) } & x_{a1} + x_{b1} \geq 40 \\
\text{d2) } & x_{a2} + x_{b2} \geq 35 \\
\text{d3) } & x_{a3} + x_{b3} \geq 25 \\
& x_{ij} \geq 0
\end{aligned}$$

2.2.4 Transformations of elements of a LP

Transforming the objective function of a linear program is straightforward. A MAX problem can be transformed into MIN (and vice versa) changing the sign of the cost coefficients:

$$\text{MIN } z = \mathbf{c}'\mathbf{x} \Leftrightarrow \text{MAX } z' = -\mathbf{c}'\mathbf{x}$$

Nonequality constraints can be transformed changing the signs of all terms of the constraint:

$$a_{i1}x_1 + \cdots + a_{in}x_n \leq b_i \Leftrightarrow -a_{i1}x_1 - \cdots - a_{in}x_n \geq -b_i$$

A nonequality constraint can be turned into equality by adding nonnegative variables:

$$\begin{aligned} a_{i1}x_1 + \cdots + a_{in}x_n \leq b_i &\Rightarrow a_{i1}x_1 + \cdots + a_{in}x_n + s_i = b_i \\ a_{k1}x_1 + \cdots + a_{kn}x_n \geq b_k &\Rightarrow a_{k1}x_1 + \cdots + a_{kn}x_n - e_k = b_k \\ s_i \geq 0,\ e_k &\geq 0 \end{aligned}$$

Less than equal constraints are turned into equality by adding slack variables s_i, and greater than equal constraints by excess variables e_k. If the original constraints have to be maintained, both types of variables have to be nonnegative.

Finally, decision variables can also be transformed. A nonpositive variable x_i can be replaced by a nonnegative variable $x_i^{'}$ making $x_i^{'} = -x_i$. A variable unconstrained in sign x_k can be replaced by two nonnegative variables $x_k^{'},\ x_k^{''}$ by making $x_k = x_k^{'} - x_k^{''}$.

2.2.5 Turning a PL into standard form

A usual transformation of a PL model is turning all constraints into equalities adding slack and excess variables. This is required to solve the PL using any version of the simplex algorithm. For instance, the model defined in subsection 2.2.2 can be put into standard form making:

$$
\begin{aligned}
&\text{MAX } z = 20P1 + 60P2 \\
&\text{s.t. WH) } 30P1 + 20P2 + h_W = 2700 \\
&\quad \text{MH } 5P1 + 10P2 + h_M = 850 \\
&\quad \text{PM) } P1 + P2 - e_P = 95 \\
&\quad P1,\ P2,\ h_W, h_M,\ e_P \geq 0
\end{aligned}
$$

where h_W and h_M are equal to the working and machine hours, respectively, not used in the proposed solution, and e_P equals the total production made over the minimal value required of 95. Note than slack and excess variables have to be also nonnegative.

In the standard form, any constraint that was an inequality in the original form will have its corresponding slack or excess variable equal to zero when it is satisfied with the equal sign. Then we will say that this constraint is *active*. If its corresponding slack or excess variable holds with the inequality sign, its corresponding variable will be positive, and the constraint will be not active.

2.3 Solving the LP

The most extended procedure to solve the LP is the *simplex algorithm*, developed by George Bernard Dantzig in 1947. This method takes advantage of the fact that the optimum or optima of a LP can be found exploring its basic solutions. A *basic solution* of a LP in standard form of n variables and m constraints has the following properties:

- has $n - m$ *nonbasic variables* equal to zero: $\mathbf{x}_N = 0$
- has m *basic variables* greater or equal to zero: $\mathbf{x}_N \geq 0$

When one or more basic variables equal zero, the solution is called *degenerate*. The basic solutions correspond to the *vertices of the feasible region*.

The strategy of the simplex method consists in:

- Finding an initial basic solution
- Explore the basic solutions moving in the direction of maximum local increase (MAX) or decrease (MIN) of the objective function
- Stop when an optimal solution is found

The software that solves LPs uses usually the simplex algorithm, or the *revised simplex algorithm*, a variant of the original simplex algorithm that is implemented more efficiently on computers. Other algorithms exist for particular LP problems, such as the transportation or transshipment problem, or the maximum flow problem.

Another approach to solve LPs is the *interior point algorithm*, developed by Narenda Karmarkar [4]. This algorithm has been proven as particularly useful in large problems with sparse matrices. Contrarily to the simplex approach, this algorithm starts from a point inside the feasible region, and approaches the optimum iteratively.

2.4 Duality in linear programming

Let's consider a MAX linear program in its canonical form:

$$\text{MAX } z = \mathbf{c}'\mathbf{x}$$
$$\text{s. t. } \mathbf{A}\mathbf{x} \leq \mathbf{b}$$
$$\mathbf{x} \geq 0$$

The following linear program, expressed in MIN canonical form, is the *dual* of the program above, called the *primal*:

$$\text{MIN } w = \mathbf{u}'\mathbf{b}$$
$$\text{s. t. } \mathbf{u}'\mathbf{A} \geq \mathbf{c}'$$
$$\mathbf{u} \geq 0$$

Note that each variable of the dual is linked with a constraint of the primal, since both share the same b_j parameter. Accordingly, each constraint of the dual is linked to a variable of the primal, as both share the same c_i parameter.

If the linear program is not expressed in canonical form, it can be turn into canonical form using the transformations defined in section 2.2. More conveniently, the dual can be obtained applying the transformations defined in Table 2.2 for the original formulation of the model.