MA423 - Fundamentals of Operations Research: Linear Programming Lectures 1 and 2

Katerina Papadaki[1]

Academic year 2018/19

[1]London School of Economics and Political Science. Houghton Street London WC2A 2AE (k.p.papadaki@lse.ac.uk)

Contents

Chapter 1

Lecture 1: Introduction

1.1 Optimization problems

The term Mathematical Programming refers to using **mathematical** tools for optimally allocating and using limited resources when planning (**programming**) activities. Mathematical programming deals therefore with optimization problems, and is also often known with the alternative name of **Mathematical Optimisation**. An optimization problem consists in maximizing or minimizing some function – representing our criterion for how "good" a solution is, such as for example, total profit or cost, or total number of staff, time, or other resources needed to perform certain tasks – subject to a number of constraints, representing limitations on the resources or on the way these can be used. In general, a mathematical programming problem is of the following form:

$$\begin{aligned} \max\ (\text{or}\ \min) \quad & f_0(x) \\ \text{s.t.} \quad & f_1(x) \leq b_1 \\ & \vdots \\ & f_m(x) \leq b_m \\ & x \in \mathcal{D} \end{aligned} \tag{1.1}$$

where

- x is a vector of *decision variables*, $x_1, \ldots, x_n$,
- $\mathcal{D}$ is the domain of the variables,
- $f_0 : \mathcal{D} \to \mathbb{R}$ is the *objective function*,
- $f_i : \mathcal{D} \to \mathbb{R}$, $i = 1, \ldots, m$, are the functions that define the *constraints* $f_i(x) \leq b_i$, where $b_i \in \mathbb{R}$.

If the problem requires to maximize the objective function, then we say that (1.1) is a *maximization problem*, if it requires to minimize, then we say that it is a *minimization problem*.

Definition 1.1 (Feasible and optimal solutions)**.**

- **Feasible solutions** *A point* $\bar{x} \in \mathcal{D}$ *is a* feasible solution *for the optimization problem* (1.1) *if it satisfies all constraints, that is,* $f_i(\bar{x}) \leq b_i$ *for* $i = 1, \ldots, m$*.*

- **Feasible region** *The* feasible region *for the optimization problem is the set of all feasible solutions*[1]*.*

- **Optimal solutions** *An* optimal solution *for a maximization problem is a feasible solution* x^* *satisfying* $f(x^*) \geq f(x)$ *for every feasible solution* x*.*

  An optimal solution *for a minimization problem is a feasible solution* x^* *satisfying* $f(x^*) \leq f(x)$ *for every feasible solution* x*.*

The aim of mathematical programming is to develop theoretical tools and algorithms in order to solve optimization problems, where by "solving" the problem we mean finding an optimal solution, or deciding that an optimal solution does not exist.

Definition 1.2 (Infeasible and unbounded problems)**.**

- **Infeasible problems** *An optimization problem is* feasible *if it has at least one feasible solution. The problem is said to be* infeasible *otherwise.*

- **Unbounded problems***A maximization problem is* unbounded *if, for every* $\alpha \in \mathbb{R}$*, there exists a feasible solution* x *such that* $f(x) \geq \alpha$*. A minimization problem is* unbounded *if, for every* $\alpha \in \mathbb{R}$*, there exists a feasible solution* x *such that* $f(x) \leq \alpha$*. An optimization problem that is not unbounded is said to be* bounded*.*

Note that, in the situation where the optimization problem is infeasible or unbounded, it does not admit an optimal solution. Indeed, in the former case the problem has no solution at all, whereas in the second case the problem has solutions that are arbitrarily "good". We also observe that an optimization problem may

[1] The feasible region should not be confused with the domain $\mathcal{D}$ of the variables. For example, consider the problem to minimize x^2 subject to the single constraint $-\log(x) \leq 0$; the domain of the problem is $\{x : x > 0\}$ (because the log function is only defined for positive numbers), whereas the feasible region is the set $\{x : x \geq 1\}$

not admit an optimal solution, even if it is feasible and bounded. For example, consider the problem

$$\begin{aligned} \min x_1 \\ x_1 x_2 \geq 1 \\ x_1, x_2 \geq 0. \end{aligned}$$

Obviously the problem is feasible, it is bounded (since $x_1 \geq 0$, hence the objective function cannot take value less than 0), but it does not have an optimal solution. Indeed, it is clear that there are feasible solutions with objective value arbitrarily close to 0, but none can have value 0. (See Figure 1.1.)

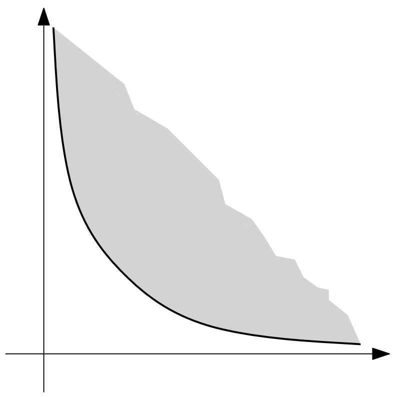


Figure 1.1: The shaded gray area represent the feasible region (which extend indefinitely).

The framework presented so far is however too general. Whether or not an optimization problem can be solved efficiently depends on the specific structure of the problem, namely the domain $\mathcal{D}$ of the decision variables and the form of the functions that define the problem. Here are three classes of optimisation problems:

Linear Programs: in this case $\mathcal{D} = \mathbb{R}^n$ and the functions $f_i : \mathbb{R}^n \to \mathbb{R}$ ($i = 0, \ldots, m$) are *linear functions*, that is, functions of the form $f(x) = \sum_{i=1}^{n} a_i x_i$ for given numbers $a_1, \ldots, a_n \in \mathbb{R}$.

Convex Programs: This is the case where the functions $f_i : \mathbb{R}^n \to \mathbb{R}$ ($i = 0, \ldots, m$) are convex.

Integer Linear Programs: they are linear programs in which some of the variables are restricted to take integer values.

We remark that, despite the formal similarities between Linear Programs and Integer Linear Programs, Integer Linear Programs are significantly harder to solve than Linear Programs.

Another vast class is that of *combinatorial optimization* problems, which are the subject of the course MA428.

1.2 Examples of Optimization Problems

Example 1.3. A factory produces orange juice (OJ) and orange concentrate (OC), starting from three raw material: electricity, oranges, and water. Every liter of OJ produced gives a profit of £3, while every liter of OC produced gives a profit of £2. Producing a liter of OJ requires 1 unit of electricity, 1 unit of oranges, and 1 unit of water. Producing a liter of OC requires 1 unit of electricity, 2 units of oranges, and it returns 1 unit of water as a byproduct of the concentration process. The factory has at its disposal 6 units of electricity, 10 units of oranges, and 4 units of water. How much OJ and OC should the factory produce in order to maximize profit?

Decision variables. We need to decide how much OJ and how much OC to produce. We introduce two variables:
x_1 = number of liters of OJ produced;
x_2 = number of liters of OC produced.

Objective function. The total profit of producing x_1 liters of OJ and x_2 liters of OC is

$$3x_1 + 2x_2.$$

Constraints. Producing x_1 liters of OJ and x_2 liters of OC we consume

$$x_1 + x_2 \text{ units of electricity,}$$

$$x_1 + 2x_2 \text{ units of oranges,}$$

$$x_1 - x_2 \text{ units of water.}$$

Since we are not allowed to use more resources than the ones at our disposal, we have the following constraints:

$$\begin{aligned} x_1 + x_2 &\leq 6 && (1.2) \\ x_1 + 2x_2 &\leq 10 && (1.3) \\ x_1 - x_2 &\leq 4 && (1.4) \end{aligned}$$

Furthermore, since the variables x_1 and x_2 represent nonnegative quantities, we need also to introduce the constraints $x_1 \geq 0$, $x_2 \geq 0$.

The problem we need to solve is therefore

$$
\begin{array}{rrcrcr}
\max & 3x_1 & + & 2x_2 & & \\
s.t. & x_1 & + & x_2 & \leq & 6 \\
 & x_1 & + & 2x_2 & \leq & 10 \\
 & x_1 & - & x_2 & \leq & 4 \\
 & & & x_1, x_2 & \geq & 0
\end{array}
$$

The above is a linear program.

Example 1.4. A hospital needs to decide how many nurses to staff. On each day i of the week ($i = 1, \ldots, 7$), the hospital requires d_i nurses (the number of nurses needed might vary from day to day). By contract, each nurse works five consecutive days and then gets two consecutive days of rest each week, but the rest days need not be the same for each nurse. How do we plan the shifts so as to minimize the total number of nurses needed?

Decision variables. In this case the choice of variables is not completely obvious. We choose the following variables: for every day i of the week, $i = 1, \ldots, 7$, we have one variable,

$$x_i = \text{ number of nurses who start working on day } i.$$

Since a nurse works 5 consecutive days and rests the remaining 2, once we have decided how many nurses start on a given day, we have completely determined the nurses' schedule.

Objective function. Each nurse we hire starts on a specific day of the week. Thus the total number of nurses hired is

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7.$$

Constraints. We need to guarantee that on any given day there is a sufficient number of nurses working. For example, the nurses working on Monday are the ones who started either on Monday, or within four days before. Thus the total number of nurses working on Monday is $x_1 + x_4 + x_5 + x_6 + x_7$. This gives the constraint

$$x_1 + x_4 + x_5 + x_6 + x_7 \geq d_1.$$

The other constraints can be derived similarly. Note also that we have also the conditions $x_i \geq 0$, $i = 1, \ldots, 7$. Another condition that we need to impose is that the number of nurses starting the shift on a given day is an integer number, we have therefore the integrality constraints:

$$x_i \text{ integer}, \; i = 1, \ldots, 7.$$

The problem we need to solve is therefore

$$
\begin{array}{rlllllllll}
\min & x_1 & +x_2 & +x_3 & +x_4 & +x_5 & +x_6 & +x_7 & & \\
s.t. & x_1 & & & +x_4 & +x_5 & +x_6 & +x_7 & \geq & d_1 \\
 & x_1 & +x_2 & & & +x_5 & +x_6 & +x_7 & \geq & d_2 \\
 & x_1 & +x_2 & +x_3 & & & +x_6 & +x_7 & \geq & d_3 \\
 & x_1 & +x_2 & +x_3 & +x_4 & & & +x_7 & \geq & d_4 \\
 & x_1 & +x_2 & +x_3 & +x_4 & +x_5 & & & \geq & d_5 \\
 & & +x_2 & +x_3 & +x_4 & +x_5 & +x_6 & & \geq & d_6 \\
 & & & +x_3 & +x_4 & +x_5 & +x_6 & +x_7 & \geq & d_7 \\
 & & \multicolumn{5}{l}{x_1, x_2, x_3, x_4, x_5, x_6, x_7} & \geq 0 & & \\
 & & \multicolumn{5}{l}{x_1, x_2, x_3, x_4, x_5, x_6, x_7} & \text{integer.} & &
\end{array}
$$

The above is an Integer Linear Program.

Example 1.5. (Markowitz portfolio optimization) We are given n assets or stocks, to be held over a given period of time. We want to decide how to allocate our budget B among the different assets during the time period. Let w_i denote the proportion of budget B invested in asset i during the time period. We assume that $\sum_{i=1}^n w_i = 1$ (i.e. we invest all our budget). Here we consider the case in which we cannot hold short positions, that is, $w_i \geq 0$, $i = 1, \ldots, n$.

The return r of the portfolio at the end of the time period is a random variable, depending on our choice of $w \in \mathbb{R}^n$. We denote by $\bar{r}$ the expected value of r. In Markowitz's model, the *risk* of the portfolio is measured as the variance of r (i.e. $\mathbb{E}[(r - \bar{r})^2]$). Given a *target return* $r_{\min}$, we want to determine the portfolio allocation vector w that minimizes risk, subject to the constraint that the expected return $\bar{r}$ is at least the target $r_{\min}$. That is, we want to determine the optimal solution for the following optimization problem.

$$
\begin{gathered}
\min \ \mathrm{Var}(r) \\
\text{s.t.} \\
\bar{r} \geq r_{\min} \\
\textstyle\sum_{i=1}^n w_i = 1 \\
w \geq 0
\end{gathered}
$$

To express $\bar{r}$ and $\mathrm{Var}(r)$ in terms of w, consider the random vector $p \in \mathbb{R}^n$, where p_i is the return of asset i at the end of the period. We denote by $\bar{p}$ the vector of means of p (i.e. $\bar{p}_i = \mathbb{E}[p_i]$), and by Σ the covariance matrix of p (i.e. $\Sigma_{ij} = \mathbb{E}[(p_i - \bar{p}_i)(p_j - \bar{p}_j)]$). (We are not concerned here on how the vector $\bar{p}$ and matrix Σ are determined, typically they are estimated based on historical data.)

Since the return of asset i is $p_i w_i$, the return of the portfolio is given by

$$r = p^\top w.$$

and therefore

$$\bar{r} = \bar{p}^\top w, \qquad \mathrm{Var}(r) = w^\top \Sigma w,$$

where you get the second one by plugging $r = p^\top w$ into $\mathrm{Var}(r) = \mathbb{E}[(r - \bar{r})^2]$ and using the formula for Σ_{ij} given above.

Thus Markowitz portfolio optimization problem is

$$\begin{gathered} \min\ w^\top \Sigma w \\ \text{s.t.} \\ \bar{p}^\top w \geq r_{\min} \\ \textstyle\sum_{i=1}^n w_i = 1 \\ w \geq 0 \end{gathered}$$

Note that all constraints are linear (and therefore convex), and that the objective function is a quadratic convex function (this can be shown because covariance matrices are always positive semidefinite). Therefore the above is an example of a convex optimization problem (indeed, it is a so called *quadratic program*, meaning that the objective function is convex quadratic and the constraints are linear).

1.3 Notations and definitions

In these notes, every vector $x \in \mathbb{R}^n$ is considered as the column vector,

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

and treated as an $n \times 1$ matrix. For any matrix M, we denote by $M^\top$ the transpose of M.

Hence, given $x, y \in \mathbb{R}^n$,

$$x^\top y = y^\top x = \sum_{j=1}^n x_j y_j = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n$$

is the *scalar product* of x and y.

A function $f : \mathbb{R}^n \to \mathbb{R}$ is *linear* if there exists a vector $a \in \mathbb{R}^n$ such that $f(x) = a^\top x$ for all $x \in \mathbb{R}^n$. The function f is said to be *affine* if there exists a vector $a \in \mathbb{R}^n$ and a number $b \in \mathbb{R}$ such that $f(x) = a^\top x + b$ for all $x \in \mathbb{R}^n$.

Given $x, y \in \mathbb{R}^n$, we will write

$$x \geq y$$

to indicate that $x_j \geq y_j$ for every $j \in \{1, \dots, n\}$; we will write

$$x > y$$

if $x_j > y_j$ for every j, $j \in \{1, \dots, n\}$; and we will write

$$x \neq y$$

if there exists $j \in \{1, \dots, n\}$ such that $x_j \neq y_j$.

We denote by 0 the zero vector, where the dimension of such vector will be clear from the context. Thus, given $x \in \mathbb{R}^n$, when we write $x \geq 0$ it will be understood that the dimensions of the vectors x and 0 are compatible, and thus 0 is the zero vector in $\mathbb{R}^n$.

1.4 References

- D. Bertsimas J.N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, (1997).

  This is an excellent modern treatment of linear programming, very clearly exposed. Also nice chapter with problem formulation examples.

- V. Chvátal, *Linear Programming*, Freeman, (1983).

  The classic textbook in linear programming. Probably the most widely read and influential text on linear programming.

- S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, (2004).

  Excellent reference for the part on Convex Optimization. Freely available online at `www.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf`.

- L.H. Wolsey, *Integer Programming*, Wiley, (1998).

  A concise first book in integer programming by one of the leading figures in the field. Several examples of problem formulations.

  A book on problem formulation.

- L. Schrage, *Optimization modeling with LINGO*, LINDO Systems Inc, 1998

  Freely available at `http://www.lindo.com/`. It is the manual of LINGO, a modeling language for linear and integer programming. The book contains many example of how to formulate problems.

Chapter 2

Lecture 1: Linear Programming Problems

2.1 Definitions and Fundamental Theorem

A *Linear Programming* (LP) problem consists of finding the maximum (resp. minimum) of a linear function subject to linear equations or inequality constraints. Thus, a linear programming problem can be written in the form

$$\begin{array}{llll} \max(\text{resp.}\min) & c^\top x & & \\ & a_i^\top x & = & b_i \quad i = 1, \dots, k \\ & a_i^\top x & \leq & b_i \quad i = k+1, \dots, r \\ & a_i^\top x & \geq & b_i \quad i = r+1, \dots, m \end{array} \tag{2.1}$$

where $b_i \in \mathbb{R}$, $i = 1, \dots, m$, $c, a_i \in \mathbb{R}^n$, $i = 1, \dots, m$, and x is a vector of variables in $\mathbb{R}^n$.

Linear programming problems are the most basic type of optimization problems, they have a rich theory and several nice properties that allow to solve them efficiently. A first such property is that, unlike general optimization problems, Linear Programs always admit an optimal solution whenever they are feasible and bounded, as we express formally in the following theorem.

Theorem 2.1 (Fundamental Theorem of Linear Programming)**.** *For any linear programming problem, exactly one of the following holds.*

1. *The problem has a finite optimum;*
2. *The Problem is infeasible;*
3. *The problem is unbounded.*

We do not prove the theorem at this stage. The validity of the statement will actually follow from the *simplex algorithm* which is probably the most used and widely known method to solve LP problems, and will be presented later in this course. As we will see, the algorithm terminates when it determines that one of the three outcomes of Theorem 2.1 holds.

We illustrate the three possible outcomes outlined in Theorem 2.1 in the next three examples.

Example (An LP with optimal solution). Consider the LP problem from the previous chapter, on deciding how much Orange Juice and Orange Concentrate to manufacture.

$$\begin{array}{rrcrcr} \max & 3x_1 & + & 2x_2 & & \\ s.t. & x_1 & + & x_2 & \leq & 6 \\ & x_1 & + & 2x_2 & \leq & 10 \\ & x_1 & - & x_2 & \leq & 4 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

The feasible region and the objective function are depicted in Figure 2.1. It can be seen that the optimum is the point $x^* = (5, 1)$, with objective value 17.

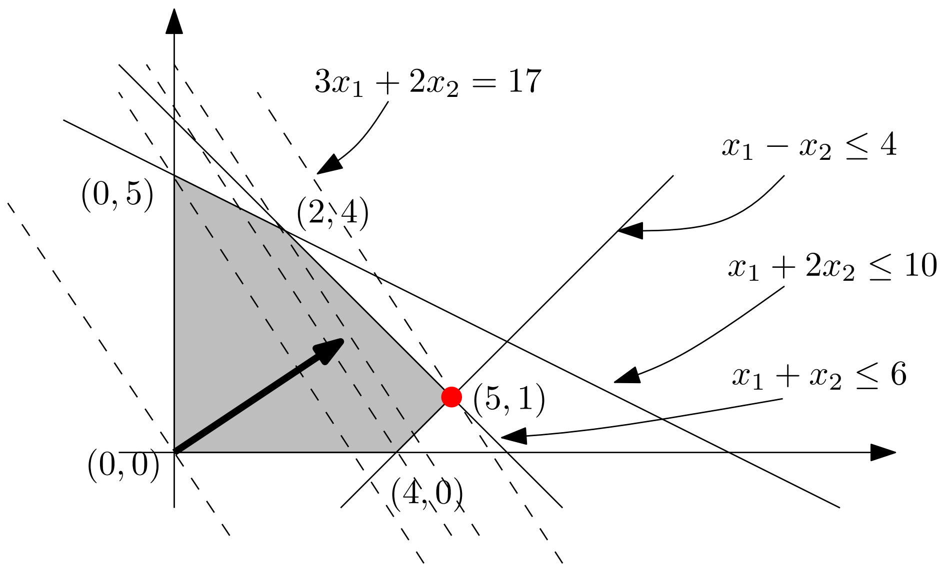


Figure 2.1: The shaded gray region is the feasible region of the problem. The boldfaced arrow represent the direction of improvement of the objective function. All points in the same dashed line have the same objective value.

Example (An infeasible LP). Consider the following LP problem.

$$
\begin{array}{lrcrcrcr}
\max & x_1 & + & x_2 & & & & \\
s.t. & x_1 & - & 2x_2 & + & 2x_3 & \leq & 2 \\
 & -2x_1 & + & 6x_2 & - & 2x_3 & \leq & -6 \\
 & & & x_1, x_2, x_3 & & & \geq & 0
\end{array}
$$

The problem has no feasible solution. To see this, note that if we multiply the second constraint by $\frac{1}{2}$ and sum it to the first constraint, we obtain the inequality

$$x_2 + x_3 \leq -1.$$

Because the above inequality is derived by multiplying inequalities of the LP by nonnegative numbers and then summing them up, it follows that the derived inequality should be satisfied by every feasible solution. However, since $x_2, x_3 \geq 0$ for every feasible solution, it must be the case that

$$x_2 + x_3 \geq 0$$

holds for every feasible solution. Clearly the two inequalities can never be satisfied both at the same time. We thus conclude that there cannot be any feasible solution.

Example (An unbounded LP). Consider the following LP problem.

$$
\begin{array}{lrcrcr}
\max & 2x_1 & - & 3x_2 & & \\
s.t. & x_1 & + & x_2 & \geq & 1 \\
 & -x_1 & + & x_2 & \leq & 2 \\
 & x_1 & - & 2x_2 & \leq & 2 \\
 & & & x_1, x_2 & \geq & 0
\end{array}
$$

Consider the following family of solutions, defined by the parameter t.

$$
\begin{array}{rcr}
x_1(t) & = & 1 + 2t \\
x_2(t) & = & t.
\end{array}
$$

Note that, for any $t \geq 0$, the point $x(t)$ is always feasible. The objective value of $x(t)$ is

$$2(1 + 2t) - 3t = 2 + t$$

which goes to infinity as t goes to infinity. Therefore the problem is unbounded, since there are feasible solutions with arbitrarily large values.

The example is illustrated in Figure 2.2. Note that the family of solutions $x(t)$, $t \geq 0$, describes a half-line starting from the point $(1, 0)$ with direction $(2, 1)$. The half line is entirely contained in the feasible region.

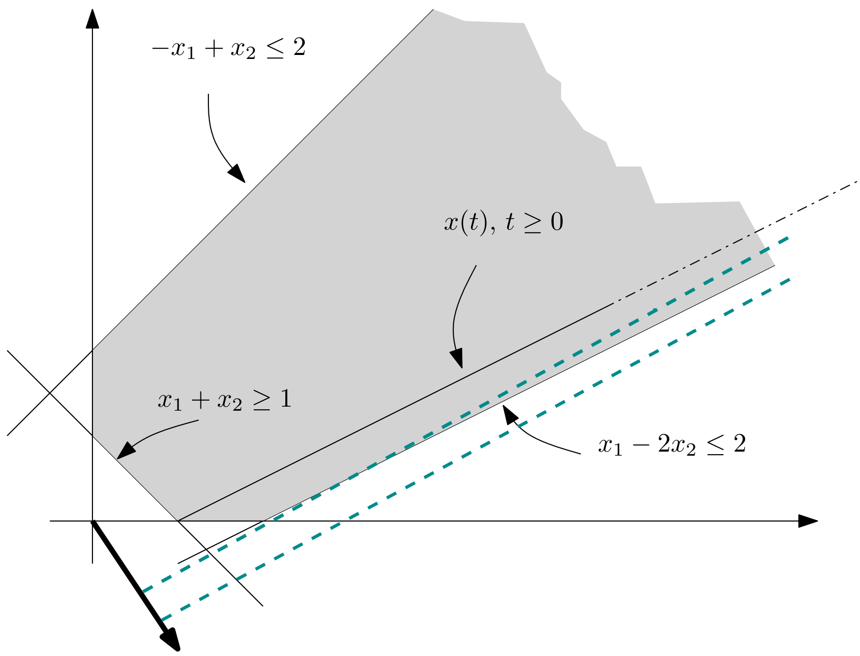


Figure 2.2: An unbounded problem.

2.2 Standard forms

When dealing with Linear Programming problems, it often convenient to assume that they are of some specific form. We will often consider problems in one of the following forms.

An LP problem is in *standard form* if it is of the form

$$\begin{aligned} z^* = \quad \max c^\top x & \\ Ax \leq & \quad b \\ x \geq & \quad 0 \end{aligned}$$

where $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and x is a vector of indeterminates in $\mathbb{R}^n$.

An LP problem is in *standard equality form* if it is of the form

$$\begin{aligned} z^* = \quad \max c^\top x & \\ Ax = & \quad b \\ x \geq & \quad 0 \end{aligned}$$

where $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and x is a vector of indeterminates in $\mathbb{R}^n$.

We want to observe here that any general LP problem (2.1) "can be brought" in one of the above standard forms, in the sense that we can always write an "equivalent " LP problem in standard form or standard equality form.

Standard form. First, observe that, if an LP problem (2.1) is a minimization problem, such problem is the same as solving $\max -c^\top x$ subject to the same constraints. We may therefore assume without loss of generality, whenever convenient, that (2.1) is a maximization problem.

"$\geq$" and "$=$" type constraints: Any "greater than or equal" constraint

$$a_i^\top x \geq b_i$$

is equivalent to the constraint

$$-a_i^\top x \leq -b_i$$

(they have the same set of feasible solutions). Similarly, any equality constraint

$$a_i^\top x = b_i$$

is equivalent to the two constraints

$$a_i^\top x \leq b_i, \quad -a_i^\top x \leq -b_i.$$

Hence we can assume that all constraints of (2.1) are of the "less than or equal" type, $a_i^\top x \leq b_i$,.

Nonpositive and free variables: If the problem has a constraint of the type $x_i \geq 0$, the variable x_i is said a *nonnegative variable*. If the problem has a constraint of the type $x_i \leq 0$, the variable x_i is said a *nonpositive variable*. Otherwise x_i is said to be a *free variable*.

- Suppose the problem has a nonpositive variable x_i. We introduce a new nonnegative variable, x_i' and we set $x_i' = -x_i$. Substituting x_i with $-x_i'$ in the constraints and in the objective function of (2.1), we get a new problem where variable x_i does not appear, but we have one new nonnegative variable. Clearly the problem we get is equivalent.

- Suppose the problem has a free variable x_i. We perform a change of variable based on the simple observation that any number can be written as the difference of two nonnegative numbers (for example, $-3 = 0 - 3$, or also

$-3 = 2 - 5$). We introduce two nonnegative variables, x_i^+ and x_i^-, and we set

$$x_i = x_i^+ - x_i^-, \quad x_i^+, x_i^- \geq 0.$$

Substituting x_i with $x_i^+ - x_i^-$ in the constraints and in the objective function of (2.1), we get a new problem where variable x_i does not appear, but we have two new nonnegative variables. Clearly the problem we get is equivalent: for any feasible solution of the new problem, we can construct a feasible solution of (2.1) with the same objective value by setting $x_i = x_i^+ - x_i^-$. Viceversa, given a feasible solution of (2.1), we get a feasible solution of the new problem with the same objective value by setting $x_i^+ = \max\{0, x_i\}$ and $x_i^- = \max\{0, -x_i\}$. For example, if $x_i = 3$, then $x_i^+ = 3$ and $x_i^- = 0$, and if $x_i = -2$ then $x_i^+ = 0$ and $x_i^- = 2$.

Standard equality form. We have seen above that any problem can be written in standard form. To transform a problem in standard equality form we have to "get rid" of all the "less than or equal" constraints, and substitute them with equality constraints. If the problem has a constraint of the form

$$a_i^\top x \leq b_i,$$

we can introduce a new nonnegative variable s_i and replace the previous constraint with the two constraints

$$a_i^\top x + s_i = b_i, \quad s_i \geq 0.$$

The variable s_i is said a *slack variable*, since $s_i = b_i - a_i^\top x$ represent the slack of the constraint $a_i^\top x \leq b_i$. Obviously, x satisfies $a_i^\top x \leq b_i$, if and only if $s_i = b_i - a_i^\top x$ satisfies $s_i \geq 0$. Since the objective function remains the same, the two problems are equivalent.

Example. Consider the following LP.

$$\begin{array}{rrcrcrcr}
\min & 3x_1 & - & 2x_2 & - & x_3 & & \\
 & -x_1 & + & x_2 & + & 2x_3 & = & 4 \\
 & 2x_1 & & & - & x_3 & \geq & -2 \\
 & -2x_1 & - & x_2 & + & x_3 & \leq & 1 \\
 & \multicolumn{7}{c}{x_1 \leq 0,\ x_3 \geq 0}
\end{array}$$

Using the procedure described above, the problem can be brought into standard from as follows:

$$\begin{array}{lllllll}
\max & 3x'_1 & +2x_2^+ & -2x_2^- & +\ x_3 & & \\
& x'_1 & +\ x_2^+ & -\ x_2^- & +2x_3 & \leq & 4 \\
& -x'_1 & -\ x_2^+ & +\ x_2^- & -2x_3 & \leq & -4 \\
& 2x'_1 & & & +\ x_3 & \leq & 2 \\
& 2x'_1 & -\ x_2^+ & +\ x_2^- & +\ x_3 & \leq & 1 \\
& & x'_1,\ x_2^+,\ x_2^-,\ x_3 \geq 0 & & & &
\end{array}$$

The optimal solution of the above problem (computed using an LP solver) is $x'_1 = 1$, $x_2^+ = 3$, $x_2^- = 0$, $x_3 = 0$, with value 9. This means that the original problem has optimal value -9, and the optimal solution is $x_1 = -x'_1 = -1$, $x_2 = x_2^+ - x_2^- = 3$, $x_3 = 0$.

The problem can be then brought into standard equality form by introducing slack variables:

$$\begin{array}{lllllllllll}
\max & 3x'_1 & +2x_2^+ & -2x_2^- & +\ x_3 & & & & & & \\
& x'_1 & +\ x_2^+ & -\ x_2^- & +2x_3 & +s_1 & & & & = & 4 \\
& -x'_1 & -\ x_2^+ & +\ x_2^- & -2x_3 & & +s_2 & & & = & -4 \\
& 2x'_1 & & & +\ x_3 & & & +s_3 & & = & 2 \\
& 2x'_1 & -\ x_2^+ & +\ x_2^- & +\ x_3 & & & & +s_4 & = & 1 \\
& & x'_1,\ x_2^+,\ x_2^-,\ x_3,\ s_1,\ s_2,\ s_3,\ s_4 \geq 0 & & & & & & & &
\end{array} \tag{2.2}$$

Incidentally, observe that, if the purpose was to bring the problem into standard equality form, we could have avoided the step to turn the equality constraint in the original problem into two "$\leq$" constraints, and simply write the problem in the form

$$\begin{array}{lllllllll}
\max & 3x'_1 & +2x_2^+ & -2x_2^- & +\ x_3 & & & & \\
& x'_1 & +\ x_2^+ & -\ x_2^- & +2x_3 & & & = & 4 \\
& 2x'_1 & & & +\ x_3 & +s_1 & & = & 2 \\
& 2x'_1 & -\ x_2^+ & +\ x_2^- & +\ x_3 & & +s_2 & = & 1 \\
& & x'_1,\ x_2^+,\ x_2^-,\ x_3,\ s_1,\ s_2 \geq 0 & & & & & &
\end{array} \tag{2.3}$$

Note that from the first two constraints of (2.2) we get that $s_1 = -s_2$ and since $s_1, s_2 \geq 0$, we must have $s_1 = s_2 = 0$. Substituting this into the first two constraints of (2.2) we get the first constraint of (2.3). Thus the formulations (2.2) and (2.3) are identical.

Chapter 3

Lecture 2: Linear Programming Duality

Ideally, given an LP problem one would like to be able to find an optimal solution. Here we address a more basic question: given an LP problem and a feasible solution, how can decide that the solution is optimal? The answer to such a question is of paramount importance when designing a method to solve LP problems, since in order to find an optimal solution we should first be able to recognize one.

Let us consider the usual example (in standard form):

$$\begin{array}{lrcr} \max & 3x_1 + 2x_2 & & \\ s.t. & x_1 + \ x_2 & \leq & 6 \\ & x_1 + 2x_2 & \leq & 10 \\ & x_1 - \ x_2 & \leq & 4 \\ & x_1, x_2 & \geq & 0 \end{array}$$

We have seen graphically that the optimal solution is $(5, 1)$, with value 17. How can we convince ourself that $(5, 1)$ is indeed the optimum? We would need to show that any feasible solution cannot have value greater than 17.

For example, if we multiply the first constraint by 3, we obtain the inequality $3x_1 + 3x_2 \leq 18$. This inequality must be satisfied by every feasible solution. Furthermore, since $x_1, x_2 \geq 0$ for every feasible solution, we have that

$$3x_1 + 2x_2 \leq 3x_1 + 3x_2 \leq 18$$

for every feasible solution. This shows that the objective value of any feasible solution x, that is $3x_1+2x_2$, cannot exceed 18, therefore the optimum value cannot be more than 18.

This does not yet prove optimality. Let us now, instead, multiply the first constraint by $\frac{5}{2}$ and the third by $\frac{1}{2}$, and let us sum them. We obtain the inequality

$$\frac{5}{2}(x_1 + x_2) \leq 15$$

and

$$\frac{1}{2}(x_1 - x_2) \leq 2.$$

Both inequalities are satisfied by any feasible solution. If we sum them, we obtain the inequality

$$3x_1 + 2x_2 \leq 17,$$

which is satisfied by every feasible solution. Note that $3x_1 + 2x_2$ is precisely the objective function, thus no feasible solution can have value more than 17. But the solution $(5, 1)$ has value exactly 17, therefore it is an optimal solution.

We will see that, according to the theory of duality, it is always possible to prove optimality with this kind of arguments.

Let us first slightly abstract the method we used in the example. Choose nonnegative multipliers for the three first constraints, say y_1, y_2, and y_3. Because $y_1, y_2, y_3 \geq 0$, the inequalities

$$\begin{array}{rcr} y_1(x_1 + x_2) & \leq & 6y_1 \\ y_2(x_1 + 2x_2) & \leq & 10y_2 \\ y_3(x_1 - x_2) & \leq & 4y_3 \end{array}$$

are satisfied by all feasible solutions x. If we sum them up, we obtain the inequality

$$(y_1 + y_2 + y_3)x_1 + (y_1 + 2y_2 - y_3)x_2 \leq 6y_1 + 10y_2 + 4y_3,$$

which is also satisfied by every feasible solution x.

Suppose now that the multipliers y_1, y_2, y_3 have been chosen so that the coefficients of the variables in the above inequality are greater than or equal to the coefficients of the variables in the objective function, which is $3x_1 + 2x_2$. That is, suppose that

$$\begin{array}{rcl} y_1 + y_2 + y_3 & \geq & 3 \\ y_1 + 2y_2 - y_3 & \geq & 2 \end{array} \tag{3.1}$$

Since $x_1, x_2 \geq 0$ for every feasible solution, we have that

$$3x_1 + 2x_2 \leq (y_1 + y_2 + y_3)x_1 + (y_1 + 2y_2 - y_3)x_2 \leq 6y_1 + 10y_2 + 4y_3$$

for every feasible solution. In this way, we would have shown that no feasible solution has value greater than $6y_1 + 10y_2 + 4y_3$. That is, any choice of nonnegative values y_1, y_2, y_3 satisfying (3.1) provides us with an upper bound on the optimal value of the LP, namely the value $6y_1 + 10y_2 + 4y_3$. The "best" upper bound we can find in this way is the optimal value of the problem

$$\begin{array}{rrrrcl} \min & 6y_1 + & 10y_2 + & 4y_3 & & \\ & y_1 + & y_2 + & y_3 & \geq & 3 \\ & y_1 + & 2y_2 - & y_3 & \geq & 2 \\ & & & y_1, y_2, y_3 & \geq & 0 \end{array}$$

The above is also a linear programming problem, called the *dual* of the original LP. In the next section we generalise the above argument to derive the dual of problems in standard form.

3.1 Duality in standard form

Consider the LP problem in standard form.

$$\begin{array}{rrl} z^* = & \max c^\top x & \\ & Ax \leq & b \\ & x \geq & 0 \end{array} \qquad (3.2)$$

where $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and x is a vector of indeterminates in $\mathbb{R}^n$.

How could we prove that a certain feasible solution $\bar{x}$ is optimal? Let $V := c^\top \bar{x}$ be the objective value of $\bar{x}$. Suppose we could somehow prove that every feasible solution x satisfies the inequality $c^\top x \leq V$. This is equivalent to saying that no feasible solution of (3.2) can have value more than V, thus we would have proven that $\bar{x}$ is indeed the optimal solution.

Thus, we want to derive upper-bounds on the optimal value z^* of (3.2). In the previous section, this was accomplished as follows: we multiplied each inequality $a_i^\top x \leq b_i$ of $Ax \leq b$ by a nonnegative multiplier y_i ($i = 1, \ldots, m$), and then summed them up to obtain a new inequality.

In matrix notation, given a nonnegative vector $y \in \mathbb{R}^m$ of multipliers, any feasible solution x for (3.2) must satisfy the inequality

$$(y^\top A)x \leq y^\top b,$$

because it is obtained by multiplying each of the constraints in $Ax \leq b$ by a nonnegative number and then summing them up. Furthermore, assume that the vector y satisfies $A^\top y \geq c$. Since x is nonnegative, x satisfies

$$c^\top x \leq (y^\top A)x \leq y^\top b = b^\top y,$$

where the second inequality holds because y is nonnegative.

Hence, given any $y \in \mathbb{R}^m$ satisfying $A^\top y \geq c$ and $y \geq 0$, for every feasible solution x for (3.2) we have $c^\top x \leq b^\top y$, thus

$$z^* \leq b^\top y.$$

In other words, $b^\top y$ is an upper-bound to the optimal value z^*. What is "the best" (i.e. the tightest) possible upper bound that we can obtain with such technique? Clearly it is the smallest possible such value. Hence, we want to find a vector y satisfying $A^\top y \geq c$, $y \geq 0$ such that $b^\top y$ is as small as possible. This is, again, a linear programming problem!

Namely, we want to solve

$$\begin{aligned} d^* = \quad & \min b^\top y \\ & A^\top y \geq \; c. \\ & \;\; y \geq 0 \end{aligned} \tag{3.3}$$

Problem (3.3) is said the *dual* of (3.2). The multipliers $y \in \mathbb{R}_m$ are the variables of the dual problem, and are called *dual variables*. We will refer to (3.2) as the *primal problem*, and the variables x of the primal problem are the *primal variables*.

We have shown the following.

Theorem 3.1 (Weak duality theorem)**.** *Given any feasible solution x^* for the primal LP (3.2) and a feasible solution y^* for its dual (3.3), we have*

$$c^\top x^* \leq b^\top y^*.$$

This immediately implies the two following corollaries.

Corollary 3.2. *Let x^* be a feasible solution for (3.2) and y^* be a feasible solution for (3.3). If $c^\top x^* = b^\top y^*$, then x^* is an optimal solution for (3.2) and y^* is optimal for (3.3).*

Corollary 3.3. *Consider the primal problem* (3.2) *and its dual* (3.3)*.*

(*i*) *If the primal problem is unbounded, then the dual is infeasible.*

(*ii*) *If the dual problem is unbounded, then the primal is infeasible.*

Proof: (*i*) Suppose the dual (3.3) has a feasible solution $\bar{y}$. Then by the weak duality theorem $c^\top x \leq b^\top \bar{y}$ for any feasible solution x for (3.2), contradicting the fact that (3.2) is unbounded. Now, suppose that the primal has a feasible solution $\bar{x}$, then by the weak duality theorem $c^\top \bar{x} \leq b^\top y$ for any feasible solution y for (3.3) contradicting the fact that (3.3) is unbounded. □

So far we have only shown that the optimal value d^* of the dual is an upper-bound to the optimal primal value z^*. When do the two values coincide? The answer is somewhat surprising: always! This is known as the *strong duality* theorem.

Theorem 3.4 (Strong duality theorem)**.** *If (3.2) has a finite optimum x^*, then also (3.3) has a finite optimum y^*, and $c^\top x^* = b^\top y^*$.*

One important consequence of the Strong Duality Theorem is the following. If a feasible solution x^* is optimal for (3.2), one can give a **certificate** of this fact; namely, an optimal solution y^* for the dual (3.3). Once given such a certificate, one can verify that x^* is optimal by

- Checking that y^* is a feasible solution for (3.3);
- Checking that $c^\top x^* = b^\top y^*$.

Example. The dual of the (by now familiar) LP

$$\begin{array}{rrcrcr} \max & 3x_1 & + & 2x_2 & & \\ s.t. & x_1 & + & x_2 & \leq & 6 \\ & x_1 & + & 2x_2 & \leq & 10 \\ & x_1 & - & x_2 & \leq & 4 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

is

$$\begin{array}{rrcrcrcr} \min & 6y_1 & + & 10y_2 & + & 4y_3 & & \\ & y_1 & + & y_2 & + & y_3 & \geq & 3 \\ & y_1 & + & 2y_2 & - & y_3 & \geq & 2 \\ & & & & & y_1, y_2, y_3 & \geq & 0 \end{array}$$

As we have seen, the optimal solution for the primal problem is $x^* = (5, 1)$ with value 17. From the discussion at the beginning of the chapter, we can argue that the optimal solution for the dual is the vector $y^* = (\frac{5}{2}, 0, \frac{1}{2})$. Note that y^* is feasible and has value 17.

3.2 Dual of the dual

What is the dual of the dual? Let us consider the problem in standard form (3.2) and its dual (3.3). To compute the dual of (3.3), we write the problem again in standard form:

$$\begin{array}{rr} -d^* = & \max -b^\top y \\ & -A^\top y \leq \ -c \ . \\ & y \geq \ 0 \end{array}$$

Then its dual is

$$\begin{aligned} \min\ & -c^\top z \\ -Az \geq & \ -b. \\ z \geq & \ 0 \end{aligned}$$

Writing the problem in standard form again, we get the equivalent problem

$$\begin{aligned} \max\ & c^\top z \\ Az \leq & \ b. \\ z \geq & \ 0 \end{aligned}$$

which is identical to (3.2).

Therefore **the dual of the dual is the primal!**

In particular, this implies the following, stronger version of the LP strong duality theorem.

Theorem 3.5 (Strong Duality Theorem)**.** *If one of the problems (3.2) and (3.3) has a finite optimum, then both problems have a finite optimum. Given an optimal solution x^* for (3.2) and an optimal solution y^* for (3.3), then $c^\top x^* = b^\top y^*$.*

The above theorem, the fundamental theorem of LP (Theorem 2.1), and Corollary 3.3, imply that any primal/dual pair of LP problems satisfies one of the alternatives represented in the table below.

		Primal		
		Fin. opt.	Infeasible	Unbounded
Dual	Fin. opt.	*Possible*	*NO*	*NO*
	Infeasible	*NO*	*Possible*	*Possible*
	Unbounded	*NO*	*Possible*	*NO*

We point out that it is indeed possible that both the primal and the dual problem are infeasible. For example, consider the following pair of primal/dual problems.

$$\begin{array}{lrcrcr} \max & 2x_1 & - & x_2 & & \\ & x_1 & - & x_2 & \leq & 1 \\ & -x_1 & + & x_2 & \leq & -2 \\ & \multicolumn{5}{r}{x_1, x_2 \geq 0} \end{array} \quad , \quad \begin{array}{lrcrcr} \min & y_1 & - & 2y_2 & & \\ & y_1 & - & y_2 & \geq & 2 \\ & -y_1 & + & y_2 & \geq & -1 \\ & \multicolumn{5}{r}{y_1, y_2 \geq 0} \end{array}$$

Observe that both problems are infeasible. Indeed, if we sum the first two constraints of the primal, we obtain the inequality $0 \leq -1$, which is never verified; if we sum the first two constraints of the dual, we obtain the inequality $0 \geq 1$.

3.3 Duality in other forms

3.3.1 Duality in standard equality form

The dual is defined for any LP problem, not just for standard form LPs. In principle, we could convert any LP problem in standard form and then compute the dual, but this is often cumbersome. We will give a general way of constructing the dual in the next section, but for now we derive the dual for problems in standard equality form.

Consider the following LP problem in standard equality form.

$$\begin{aligned} z^* = \quad & \max c^\top x \\ Ax = \quad & b \\ x \geq \quad & 0 \end{aligned} \tag{3.4}$$

where $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and x is a vector of indeterminates in $\mathbb{R}^n$.

What is the dual of (3.4)? We have two way of finding the dual: one is to repeat the argument we did for LPs earlier in standard form, with the necessary modifications. The other is to write (3.4) in standard form and then use the form of the dual given in (3.3).

Method 1 As before, we look for upper bounds to the optimal value z^* of (3.4). Note that any vector x satisfying $Ax = b$ also satisfies

$$(y^\top A)x = y^\top b,$$

for any vector $y \in \mathbb{R}^m$, but note that here y need not be nonnegative. Furthermore, suppose that y also satisfies $A^\top y \geq c$. Then, for any feasible solution x for (3.4), x satisfies the inequality

$$c^\top x \leq (y^\top A)x$$

because x is nonnegative. Combining the two above inequalities, we obtain, for any feasible solution x for (3.4) and any $y \in R^m$ such that $A^\top y \geq c$, the following inequality

$$c^\top x \leq b^\top y.$$

Thus any $y \in \mathbb{R}^m$ satisfying $A^\top y \geq c$, provides an upper-bound to the optimal value z^* of (3.4), namely

$$z^* \leq b^\top y.$$

The best possible upper-bound we can obtain in this way is the optimal value d^* of the LP problem

$$\begin{aligned} d^* = \quad & \min b^\top y \\ A^\top y \geq \quad & c. \end{aligned} \tag{3.5}$$

The problem (3.5) is the dual of (3.4).

Method 2 Let us write (3.4) in standard form by replacing each equality constraint $a_i^\top x = b_i$ with two inequality constraints $a_i^\top x \leq b_i$ and $-a_i^\top x \leq -b_i$. We get the problem

$$\begin{aligned} z^* = \quad & \max c^\top x \\ Ax \leq \quad & b \\ -Ax \leq \quad & -b \\ x \geq \quad & 0 \end{aligned}$$

Using (3.2), we can write the dual of the above problem:

$$\begin{aligned} \min b^\top y^+ - b^\top y^- \\ A^\top y^+ - A^\top y^- \geq c \\ y^+, y^- \geq 0. \end{aligned}$$

If we make a change of variables $y = y^+ - y^-$, y is now a vector of free variables in $\mathbb{R}^m$. Note that we get exactly (3.5).

3.3.2 Duality for general problems

So far we have derived the dual problem only for problems in standard equality form. One can, in fact, associate a dual problem to any linear programming problem, whether it is a maximization or minimization problem, regardlessly of the particular form of the linear constraints.

Consider an $m \times n$ matrix A with rows $a_1^\top, \ldots, a_m^\top \in \mathbb{R}^n$ and columns $A_1, \ldots, A_n \in \mathbb{R}^m$, and let $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$.

If the primal is a maximization problem, then the dual is a minimization problem, and viceversa. The dual problem will have a variable for every constraint of the primal, except for the nonnegativity or nonpositivity constraint, and a constraint for every variable of the primal. The coefficient of a primal variable in the objective function will be the right-hand-side of the corresponding dual constraint, and the right-hand-side of a primal constraint will be the objective function coefficient of the corresponding dual variable.

The "recipe" to go from to construct the dual of an LP problem is given by the following table.

$\max c^\top x$	$\min b^\top y$	
$a_i^\top x \le b_i,$	$y_i \ge 0,$	$i = 1, \dots, h;$
$a_i^\top x \ge b_i,$	$y_i \le 0,$	$i = h+1, \dots, k;$
$a_i^\top x = b_i,$	y_i free,	$i = k+1, \dots, m.$
$x_j \ge 0,$	${A_j}^\top y \ge c_j,$	$j = 1, \dots, p;$
$x_j \le 0,$	${A_j}^\top y \le c_j,$	$j = p+1, \dots, q;$
x_j free	${A_j}^\top y = c_j,$	$j = q+1, \dots, n;$

The above table can be read in both directions: from left to right if we want to write the dual of a maximization problem, from right to left if we want to write the dual of a minimization problem.

For example, in a maximization problem, constraints of the type "$\geq$" correspond dual variables restricted to be "≤ 0". In a minimization problem, instead, to constraints of the type "$\geq$" correspond dual variables restricted to be "≥ 0". Note that equality constraints in the primal correspond to free variables in the dual, and free variables in the primal correspond to equality constraints in the dual.

Example. Consider the LP in the example of Section 2.2

$$
\begin{array}{rlcr}
\min & 3x_1 - 2x_2 - x_3 & & \\
 & -x_1 + x_2 + 2x_3 & = & 4 \\
 & 2x_1 - x_3 & \geq & -2 \\
 & -2x_1 - x_2 + x_3 & \leq & 1 \\
 & x_1 \leq 0,\ x_3 \geq 0 & &
\end{array}
$$

The problem is a minimization problem, thus the dual is a maximization problem. The problem has three constraints (besides the nonpositivity constraint $x_1 \leq 0$ and the nonnegativity constraint $x_3 \geq 0$), thus the dual will have three variables, y_1, y_2, y_3. The problem has three variables, thus the dual will have three constraints (besides the nonnegativity and nonpositivity constraints on some of the variables). The dual is the following

$$
\begin{array}{rlcr}
\max & 4y_1 - 2y_2 + y_3 & & \\
 & -y_1 + 2y_2 - 2y_3 & \geq & 3 \\
 & y_1 - y_3 & = & -2 \\
 & 2y_1 - y_2 + y_3 & \leq & -1 \\
 & y_2 \geq 0,\ y_3 \leq 0 & &
\end{array}
$$

One can verify that, if we write the dual of the above problem, we end up with the original one.

3.4 An economics interpretation of the dual

When one formulates a real-world problem as an LP, the variables, constraints, and objective function of the LP typically have a clear interpretation in terms of the original problem. It is often the case that also the variables, constraints, and objective function of the dual can be interpreted and give further information on the original problem. A typical case when such an interpretation of the dual is possible is in the case of resource allocation problems. Here we give an example of such a problem.

A chip's manufacturer produces four types of memory chips in one their state-of-the-art factories. These chips are sold to electronics corporations for their devices. The main resources used in the chip's production are labor and silicon wafers.

The factory's problem for the next month is

$$\begin{array}{lrcll} \text{maximise} & 15x_1 + 24x_2 + 32x_3 + 40x_4 & & & \\ \text{subject to} & x_1 + \ 2x_2 + \ 8x_3 + \ 7x_4 & \leq & 2000 & \text{(Labour)} \\ & 6x_1 + \ 8x_2 + 12x_3 + 15x_4 & \leq & 15000 & \text{(Silicon wafers)} \\ & x_1,\, x_2,\, x_3,\, x_4 \geq 0. & & & \end{array}$$

where variable x_j represents the number of chips produced, in thousands, the objective function coefficients represent the unit profit (in £) of the four types of chips, 2000 is the number of hours of labour available, and 15000 is the number of silicon wafers available. Therefore, the objective function's coefficients represent thousand £ per unit of each variable $x_1, \ldots, x_4$ (so, for example, $x_2 = 1$ corresponds to producing 1000 chips of type 2, which requires 2 hours of labour, 8 wafers, and gives a profit of £24000, that is, $24x_2$ thousand £.)

Mango Inc., a giant consumer electronics corporation, urgently needs as many units as possible of a new type of memory chip, due to a stronger-than-expected demand for their new smart phone. Mango Inc. would like the chip manufacturer to sell all its resources (labour and wafers) to Mango for the production of this new chip. Mango Inc. intends to determine prices to offer for each of the resources in order to convince the manufacturer to sell them, while minimising the total sum paid.

Let y_1 denote the price (in thousand £) that the buyer intends to pay per hour of labour, and let y_2 denote the price (in thousand £) that Mango Inc. intends to pay for each silicon wafer.

Clearly these prices must be greater than or equal to zero, that is, $y_1, y_2 \geq 0$.

In order to persuade the manufacturer to sell, the buyer must offer the prices such that the manufacturer is not tempted to retain its resources to produce chips of type 1. So the buyer must offer prices such that the total value of the resources used in producing chips of type 1 is at least the amount of profit the manufacturer could attain.

That is

$$y_1 + 6y_2 \geq 15.$$

Similarly for chips of type 2, Mango Inc. must pitch his prices so that the manufacturer is at least as well off selling as he would be by not selling:

$$2y_1 + 8y_2 \geq 24.$$

Each type of chip generates such a constraint. On the other hand, the buyer wants to pay the minimum amount possible for the entire amount of resources, that is, it wants to minimise $2000y_1 + 15000y_2$.

Mango Inc.'s problem is therefore

$$\begin{array}{lrcl} \text{minimise} & 2000y_1 + 15000y_2 & & \\ \text{subject to} & y_1 + 6y_2 & \geq & 15 \\ & 2y_1 + 8y_2 & \geq & 24 \\ & 8y_1 + 12y_2 & \geq & 32 \\ & 7y_1 + 15y_2 & \geq & 40 \\ & y_1,\, y_2 & \geq & 0 \end{array}$$

The manufacturer's and the Mango Inc.'s problems are dual to each other. Under this interpretation, the dual values represent prices of resources, which explains why dual values are often also called *shadow prices*. Of course, we know that the total amount that the buyer will have to pay is the same as the amount of profit that the manufacturer would achieve by carrying on its usual production.

3.5 Optimality conditions

3.5.1 Standard form

Let us look more closely at the proof of the weak duality theorem (Theorem 3.1). Consider the LP problem (P) in standard form, and its dual (D).

$$\begin{array}{rll} \max \sum_{j=1}^{n} c_j x_j & & \\ \sum_{j=1}^{n} a_{ij} x_j \leq & b_i & i = 1, \ldots, m \\ x_j \geq & 0 & j = 1, \ldots, n \end{array} \quad (P) \qquad ; \qquad \begin{array}{rll} \min \sum_{i=1}^{m} b_i y_i & & \\ \sum_{i=1}^{m} a_{ij} y_i \geq & c_j & j = 1, \ldots, n \\ y_i \geq & 0 & i = 1, \ldots, m \end{array} \quad (D)$$

Let x^* and y^* be feasible solutions for (P) and (D) respectively.

We have

$$\sum_{j=1}^{n} c_j x_j^* \leq \sum_{j=1}^{n} (\sum_{i=1}^{m} a_{ij} y_i^*) x_j^* = \sum_{i=1}^{m} (\sum_{j=1}^{n} a_{ij} x_j^*) y_i^* \leq \sum_{i=1}^{m} b_i y_i^*. \tag{3.6}$$

where the first inequality follows from the facts that $\sum_{i=1}^{m} a_{ij} y_i \geq c_j$ and that $x_j \geq 0$ for $j = 1, \ldots, n$, and the second inequality follows from the facts that $\sum_{j=1}^{n} a_{ij} x_j \leq b_i$ and that $y_i \geq 0$ for $i = 1, \ldots, m$.

By strong duality, x^* and y^* are optimal for (P) and (D), respectively, if and only if $\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} b_i y_i^*$. This is the case if and only if equality holds throughout in the chain of inequalities (3.6). Hence, x^* and y^* are optimal if and only if

$$\sum_{j=1}^{n} c_j x_j^* = \sum_{j=1}^{n} (\sum_{i=1}^{m} a_{ij} y_i^*) x_j^*, \quad \text{and} \quad \sum_{i=1}^{m} (\sum_{j=1}^{n} a_{ij} x_j^*) y_i^* = \sum_{i=1}^{m} b_i y_i^*.$$

The first condition holds if and only if

$$\sum_{j=1}^{n} (\sum_{i=1}^{m} a_{ij} y_i^* - c_j) x_j^* = 0. \tag{3.7}$$

The second condition holds if and only if

$$\sum_{i=1}^{m} (\sum_{j=1}^{n} a_{ij} x_j^* - b_i) y_i^* = 0. \tag{3.8}$$

Note that, since x^* and y^* are feasible for (P) and (D), they satisfy $x_j^* \geq 0$ and $\sum_{i=1}^{m} a_{ij} y_i^* - c_j \geq 0$ for all $j = 1, \ldots, n$, and $y_i^* \geq 0$, $\sum_{j=1}^{n} a_{ij} x_j^* - b_i \leq 0$ for all $i = 1, \ldots, m$. Since all the terms of (3.7) are positive, the sum is zero if and only if each term is zero. Similarly, since all the terms of (3.8) are negative, the sum is zero if and only if each term is zero.

Therefore the last two equations hold if and only if

$$(\sum_{i=1}^{m} a_{ij} y_i^* - c_j) x_j^* = 0, \quad \text{for } j = 1, \ldots, n, \tag{3.9}$$

and

$$(\sum_{j=1}^{n} a_{ij} x_j^* - b_i) y_i^* = 0, \quad \text{for } i = 1, \ldots, m. \tag{3.10}$$

We have proved the following.

Theorem 3.6 (Complementary slackness theorem). *Given the problem* (P) *in standard form and feasible solutions* x^* *and* y^* *for* (P) *and its dual* (D)*, respectively,* x^* *and* y^* *are optimal if and only if the following* complementary slackness conditions *hold*

$$\begin{array}{lccl} \forall\, j \in \{1,\dots,n\}, & x_j^* = 0 & \textbf{or} & \sum_{i=1}^m a_{ij} y_i^* - c_j = 0, \\ \forall\, i \in \{1,\dots,m\}, & y_i^* = 0 & \textbf{or} & \sum_{j=1}^n a_{ij} x_j^* - b_i = 0. \end{array} \qquad (CS)$$

3.5.2 General form

The complementary slackness theorem holds for problems in general form.

Theorem 3.7 (Complementary slackness theorem). *Let* (P) *be a linear programming problem in the variables* $x_1,\dots,x_n$*, with* m *constraints of the form* $\sum_{j=1}^n a_{ij}x_i \le b_i$*,* $\sum_{j=1}^n a_{ij}x_i \ge b_i$*, or* $\sum_{j=1}^n a_{ij}x_i = b_i$ *(*$i = 1,\dots,m$*) plus possibly nonnegativity or nonpositivity constraints on the variables.*

Given a feasible solution x^* *for* (P) *and a feasible solution* y^* *for its dual* (D)*,* x^* *and* y^* *are optimal if and only if the following* complementary slackness conditions *hold:*

$$\begin{array}{lccl} \forall\, j \in \{1,\dots,n\}, & x_j^* = 0 & \textbf{or} & \sum_{i=1}^m a_{ij} y_i^* - c_j = 0, \\ \forall\, i \in \{1,\dots,m\}, & y_i^* = 0 & \textbf{or} & \sum_{j=1}^n a_{ij} x_j^* - b_i = 0. \end{array} \qquad (CS)$$

Observe that, if a primal variable x_j is free, then the corresponding dual constraint is the equation $\sum_{i=1}^m a_{ij}y_i - c_j = 0$, while if a dual variable y_i is free, the corresponding primal constraint is the equation $\sum_{j=1}^n a_{ij}x_i = b_i$. Therefore, the complementary slackness conditions corresponding to free variables are always trivially satisfied.

Note that the above theorem also implies the following: suppose we are given a primal feasible solution x^* and we want decide if it is optimal. By the complementary slackness theorem, this is the case if and only if there exists a dual feasible solution such that x^*, y^* satisfy the complementary slackness conditions.

Example Consider the following LP problem from the example of Section 2.2

$$\begin{array}{rllll} \min & 3x_1 & -2x_2 - & x_3 & & \\ & -x_1 & +\ x_2 + 2x_3 & = & 4 \\ & 2x_1 & \qquad - \ x_3 & \ge & -2 \\ & -2x_1 - & x_2 + \ x_3 & \le & 1 \\ & \multicolumn{4}{c}{x_1 \le 0,\ x_3 \ge 0} \end{array}$$

and its dual

$$\begin{array}{rl} \max & 4y_1 - 2y_2 + y_3 \\ & -y_1 + 2y_2 - 2y_3 \geq 3 \\ & y_1 - y_3 = -2 \\ & 2y_1 - y_2 + y_3 \leq -1 \\ & y_2 \geq 0,\ y_3 \leq 0 \end{array}$$

We have mentioned in Section 2.2, without proving it, that the optimal solution is the point $x^* = (-1, 3, 0)$. We now use the Complementary Slackness conditions to prove optimality of x^*. First, one can easily verify that x^* satisfies the primal constraints, thus it is feasible. To prove that x^* is optimal, we compute a feasible dual solution y^* that is in complementary slackness with x^*.

The complementary slackness conditions are

$$\begin{array}{lcc} x_1^* = 0 & \text{or} & -y_1^* + 2y_2^* - 2y_3^* = 3 \\ x_2^* = 0 & \text{or} & y_1^* - y_3^* = -2 \\ x_3^* = 0 & \text{or} & 2y_1^* - y_2^* + y_3^* = -1 \\ y_1^* = 0 & \text{or} & -x_1^* + x_2^* + 2x_3^* = 4 \\ y_2^* = 0 & \text{or} & 2x_1^* - x_3^* = -2 \\ y_3^* = 0 & \text{or} & -2x_1^* - x_2^* + x_3^* = 1 \end{array}$$

The third, forth and fifth conditions are already satisfied because $x_3^* = 0$ and x^* satisfies as equality the first and second primal constraints. The remaining conditions are satisfied if and only if y^* satisfies

$$\begin{array}{rl} -y_1^* + 2y_2^* - 2y_3^* = & 3 \\ y_1^* - y_3^* = & -2 \\ y_3^* = & 0 \end{array}$$

The only solution to these equations is $y^* = (-2, \frac{1}{2}, 0)$. We have to verify that y^* is indeed feasible for the dual problem, and this is easily checked to hold. Hence $(-1, 3, 0)$ is an optimal primal solution while $(-2, \frac{1}{2}, 0)$ is an optimal dual solution. Observe that, as expected in light of the Strong Duality Theorem, the value of x^* in the primal (namely, -9) equals the value of y^* in the dual.

MA423 - Fundamentals of Operations Research
Lecture 3: The Simplex Method

Katerina Papadaki[1]

Academic year 2017/18

[1]London School of Economics and Political Science. Houghton Street London WC2A 2AE (k.p.papadaki@lse.ac.uk)

Contents

Chapter 1

Preliminaries: definitions and geometrical insights

In the last two lectures we looked at formulations of linear programs, linear programs in standard form and duality. In this lecture we will introduce the simplex method an algorithm for solving linear programs. But before we do that we need to look at some geometric concepts regarding linear programming.

1.1 Basic solutions and extreme points

Consider the LP problem

$$
\begin{array}{rrcl}
\text{maximise} & 2x_1 + 8x_2 & & \\
\text{subject to} & 2x_1 + x_2 & \leq & 10 \\
& x_1 + 2x_2 & \leq & 10 \\
& x_1 + x_2 & \leq & 6 \\
& x_1 + 3x_2 & \leq & 12 \\
& 3x_1 - x_2 & \geq & 0 \\
& x_1 + 4x_2 & \geq & 4 \\
& x_1, x_2 & \geq & 0
\end{array}
\tag{1.1}
$$

depicted graphically in Figure 1.1.

The optimal solution of the LP (1.1) is the point A of coordinates $x_1 = 1.2$, $x_2 = 3.6$. It can be observed from the diagram that this point is a corner point of the polygon that defines the feasible region. As we shall see, the fact that the optimum is achieved by some "corner point" of the feasible region is true in general, even for problems with more than two variables. Firstly, we need to define

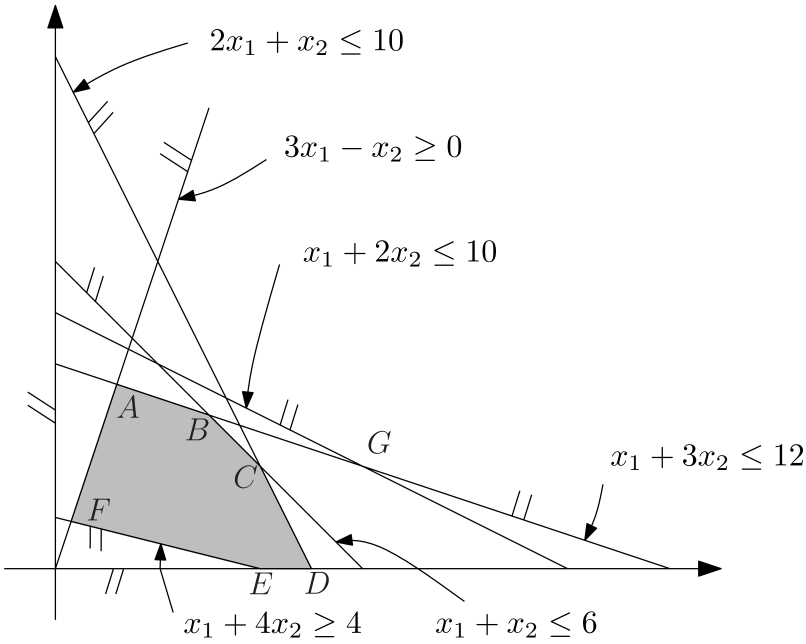


Figure 1.1: The feasible region is represented by the area shaded gray.

what we mean by "corner points", which in the language of Linear Programming are called *extreme points*.

In two dimensions a corner point is one at the intersection of two non-parallel inequalities. In order to formalise the concept to the case of more variables, we need to recall the following concept from linear algebra.

Independent constraints: Given a system of n linear inequalities in n variables, we say that the n inequalities are *(linearly) independent* if there exists a unique solution satisfying all n of them at equality.

For example, the inequalities

$$\begin{array}{rrrcl} x_1 & +x_2 & & \leq & 2 \\ x_1 & & +x_3 & = & 2 \\ & x_2 & +x_3 & \geq & 2 \end{array}$$

are independent because the only solution satisfying all three of them at equality is the point $(x_1^*, x_2^*, x_3^*) = (1, 1, 1)$, as one can verify.

On the other hand, the inequalities

$$\begin{array}{rrrcl} x_1 & +x_2 & & \leq & 2 \\ x_1 & & +x_3 & = & 2 \\ & -x_2 & +x_3 & \geq & 0 \end{array}$$

are not independent because there are multiple solutions satisfying all three of them at equality, for example the points $(1,1,1)$ and $(0,2,2)$.

Note that in 2 dimensions two equations are independent if they do not define parallel lines. If they do define parallel lines and if the two parallel lines are distinct then there are no solutions satisfying both equations, while if the two lines coincide then there are infinitely many solutions satisfying both equations.

To simplify matters, in all our examples and in all exercises we will always work under the assumption that any n equalities we deal with are independent, unless otherwise specified.

Consider a system of linear constraints in n variables.

Basic point: A *basic point* for the system is a point satisfying at equality n independent constraints from the system.

Defining constraints (active constraints): Given a basic point $\bar{x}$ for the system, a set of *defining constraints for* $\bar{x}$ (also called *active constraints for* $\bar{x}$) is any choice of n independent constraints satisfied at equality by $\bar{x}$.

Note that there might be more than one set of defining constraints for each basic point as we will see later.

Example. Consider the LP in equation (1.1), whose feasible region is represented in Figure 1.1. The basic point labelled B is defined by the intersection of constraints 3 and 4, so it is the unique solution of the system:

$$\begin{array}{rcl} x_1 + x_2 & = & 6 \\ x_1 + 3x_2 & = & 12 \end{array}$$

Basic point D is defined by constraint 1 and the nonnegativity of x_2, that is:

$$\begin{array}{rcl} 2x_1 + x_2 & = & 10 \\ x_2 & = & 0 \end{array}$$

Note that, in the definition, we do not insist that a basic point is feasible for the whole system of constraints. For example, basic points B and D are feasible. Basic point G defined by

$$\begin{aligned} 2x_1 + x_2 &= 10 \\ x_1 + 3x_2 &= 12 \end{aligned}$$

is infeasible.

Extreme point: A point that is both basic and feasible for a given system of linear constrains.

For example, points A to F in Figure 1.1 are extreme points while points G and the origin are basic but they are not extreme.

1.1.1 Is a point an extreme point?

For a system of linear constraints in n-variables $x_1, \ldots, x_n$, to determine whether a point x^* is extreme:

- Substitute the values of $x_1^*, x_2^*, \ldots, x_n^*$ into the constraints, and verify if the point is feasible.
- Determine whether n (or more) of them are satisfied as equalities,
- Among the constraints satisfied at equality by x^*, determine if there are n of them that are independent.

For example, consider as usual the system of constraints (1.1), and let $x^* = (4, 2)$.

Substituting the values $x_1 = 4$, $x_2 = 2$ into the constraints, we obtain

$$\begin{array}{rlcll} 2 \cdot 4 + 2 & = 10 & = & 10 & \surd \\ 4 + 2 \cdot 2 & = 8 & \leq & 10 & \\ 4 + 2 & = 6 & = & 6 & \surd \\ 4 + 3 \cdot 2 & = 10 & \leq & 12 & \\ 3 \cdot 4 - 2 & = 10 & \geq & 0 & \\ 4 + 4 \cdot 2 & = 12 & \geq & 4 & \\ 4, 2 & & \geq & 0 & \end{array}$$

The point is basic as constraint 1 and constraint 3 are satisfied as equalities and they are independent. The point (4, 2) is point C in the diagram in Figure 1.1. Point C is an extreme point as these calculations show that it is feasible as well as basic.

Note: In general, a basic point might have more than one set of defining constraints. For example, point $\bar{x} = (1, 1, 1)$ is basic for the following system

$$\begin{array}{cccccc} x_1 & +x_2 & & \leq & 2 \\ x_1 & & +x_3 & = & 2 \\ & x_2 & +x_3 & \geq & 2 \\ x_1 & -x_2 & +x_3 & \geq & 1 \end{array}$$

but each subset consisting of three constraints is defining for $\bar{x}$, since $\bar{x}$ satisfies at equality all four constraints and every three of them are independent.

1.1.2 Optimality and extreme points

The following is one of the most fundamental and useful facts in Linear Programming.

Extreme optimal solutions. Whenever an LP admits an optimal solution, there exists some optimal solution that is an extreme point of the feasible region.

This fact tells us that, when solving an LP, one only needs to search among the extreme points, and pick the one with best objective value.

Note: a linear programming problem may have non-extreme points that are also optimal, but according to the statement above there is always at least an extreme point on the optimal contour. For example, consider the LP in (1.2) depicted in Figure (1.2).

$$\begin{array}{rrcl} \text{maximise} & 2x_1 + 2x_2 & & \\ \text{subject to} & x_1 + x_2 & \leq & 4 \\ & x_2 & \leq & 3 \\ & x_1, x_2 & \geq & 0 \end{array} \tag{1.2}$$

(1.2) has two extreme points that are optimal, namely the points $(1, 3)$ and $(4, 0)$, but there are also infinitely many optimal solutions that are not extreme, namely all points in the line segment between $(1, 3)$ and $(4, 0)$.

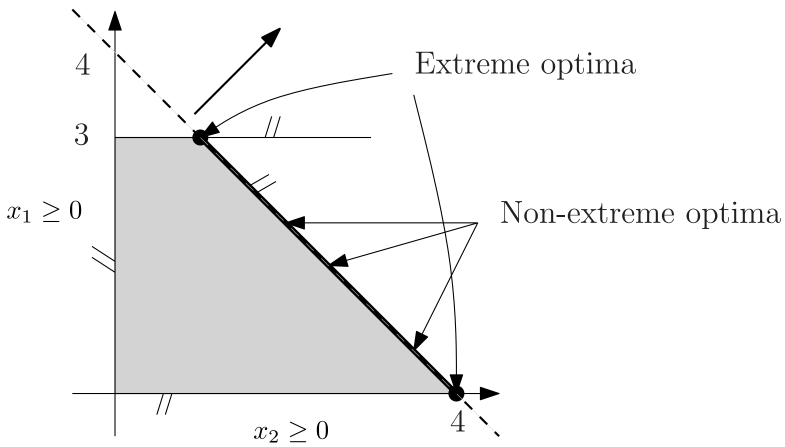


Figure 1.2: Problem (1.2) has two extreme optimal solutions and infinitely many non-extreme optimal solutions.

1.2 Effective Constraints and basic variables

Constraints of the form $x_i \geq 0$ are called *nonnegativity constraints* and all other constraints are called *resource constraints*.

We have seen that an extreme point of an LP in n variables is defined by n independent constraints satisfied at equality at the point. The constraints that define the extreme point can either belong to the resource constraints or to the nonnegativity constraints. This leads to the following definitions.

Consider an extreme point $\bar{x}$ of a system of linear constraints, and a set S of constraints defining $\bar{x}$.
Effective constraints. The resource constraints in S which are defining for $\bar{x}$ are called the *effective constraints* at $\bar{x}$ with respect to S. The remaining resource constraints are *ineffective*.

Basic variables. If a nonnegativity constraint, say constraint $x_j \geq 0$, is defining at $\bar{x}$ with respect to S, then we say that x_j is a *non-basic variable* with respect to S. The other variables are *basic variables* at that point with respect to S.

Observe that non-basic variables always have value 0. Thus, the only variables that may take positive values in the extreme point are the basic ones. However, it is possible for a basic variable to take 0 value (see section 1.4).

Given that an extreme point is defined by n independent constraints, the number of nonbasic variables plus the number of effective constraints must be n. Considering that the number of basic variables is n minus the number of nonbasic variables, it follows that:

> At any extreme point, the number of effective constraints equals the number of basic variables.

For example, consider the linear program

$$\begin{array}{rrcl} \text{maximise} & 3x_1 + 5x_2 + 2x_3 & & \\ \text{subject to} & 2x_1 + 2x_2 + x_3 & \leq & 4 \\ & 3x_1 - x_2 + 2x_3 & \leq & 5 \\ & x_1 - x_2 - x_3 & \geq & 1 \\ & x_1 \qquad\qquad & \geq & 0 \\ & x_2 \quad\;\; & \geq & 0 \\ & x_3 & \geq & 0 \end{array} \tag{1.3}$$

For the LP in (1.3), point x^* of coordinates $x_1 = 1.5$, $x_2 = 0.5$, $x_3 = 0$ has one set of three defining constraints 1, 3 and 6. Thus this is basic point and since it is also feasible this is an extreme point. Further, the effective constraints are constraints 1 and 3, variable x_3 is non-basic. Constraint 2 is ineffective and variables are x_1 and x_2 are basic. Note that the number of basic variables is equal to the number of effective constraints.

1.3 Proving Optimality Revisited

In the last lecture we created the dual problem in order to prove optimality of a feasible solution x^*. We found the dual values (positive multipliers) for all the constraints to derive an inequality with the same coefficients as the objective.

However, if x^* is a basic point and we know its n defining constraints then we only need to calculate coefficients (dual values) for the defining constraints to prove optimality. Let's see how this is done geometrically.

For example, consider the two inequalities

$$\begin{array}{rcl} 2x_1 + x_2 & \leq & 10 \\ x_1 + 3x_2 & \leq & 12 \end{array} \tag{1.4}$$

which intersect at $x_1 = 3.6$, $x_2 = 2.8$. Deriving a new constraint with arbitrarily chosen weights of 2 for the first constraint and 1 for the second gives

$$\begin{array}{rrcl} 2 \cdot (& 2x_1 + x_2 & \leq & 10) \\ 1 \cdot (& x_1 + 3x_2 & \leq & 12) \\ \hline & 5x_1 + 5x_2 & \leq & 32 \end{array}$$

The point $x_1 = 3.6$, $x_2 = 2.8$ satisfies $5x_1 + 5x_2 \leq 32$ as an equality. This relationship is illustrated in Figure 1.3.

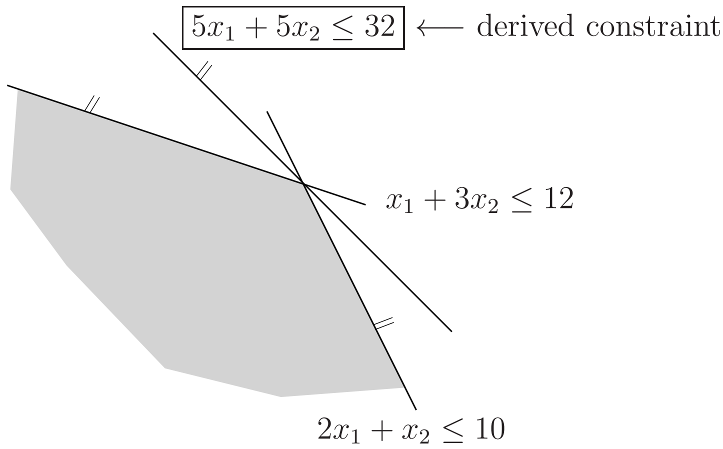


Figure 1.3: Derived constraint.

Now, consider an LP where we want to maximize $5x_1 + 5x_2$ with several constraints including the ones in (1.4) but point $(3.6, 2.8)$ satisfies only the two constraints from (1.4) at equality. So, constraints (1.4) are its defining constraints. A derived inequality such as $5x_1 + 5x_2 \leq 32$ is satisfied by all feasible solutions (since it is a non-negative linear combination of the constraints). Since the objective $5x_1 + 5x_2$ is always less than or equal to 32 and point $(3.6, 2.8)$ achieves the value of 32, this must be an optimal point.

We now have the ingredients for a test to determine whether or not a given point is optimal. Ask the question: is it possible to represent the optimal contour of the objective function as the non-negative combination of a set of defining constraints at that the point? If the answer is yes, then the point is optimal. If the answer is no and there is no other set of defining constraints then the point is not optimal. If there are other sets of defining constraints then we need to try the same thing with the other sets of defining constraints (see section 1.4).

Thus, we only need the defining constraints to prove optimality.

1.4 A degenerate example.

Consider the LP

$$\begin{array}{lrcl} \max & x_1 - x_2 & & \\ & x_1 + x_2 & \leq & 1 \\ & x_1 & \leq & 1 \\ & x_2 & \geq & 0 \end{array}$$

The problem is represented in the Figure 1.4

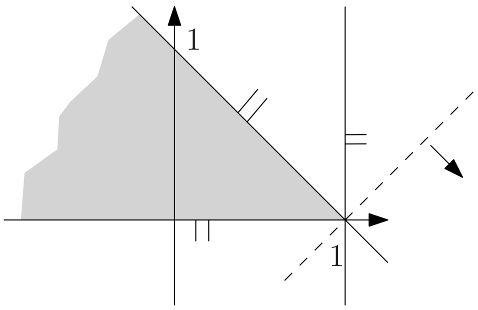


Figure 1.4: Representation of the LP (1.5). The jagged edge of the shaded area indicates that the feasible region continues indefinitely.

It is immediately clear from the figure that the optimal solution is the point $x^* = (1, 0)$, with objective value equal to 1. This problem is "degenerate" in the sense that x^* satisfies as equality three constraints, whereas the problem has only two variables. Since, there can only be two independent equalities that pass from the same point (in two dimensions), the third equality can be derived from the other two. Subtracting $x_1 + x_2 = 1$ from $x_1 = 1$ term by term gives us $x_2 = 0$. This is what makes this example to be degenerate.

Therefore there are three choices of defining constraints for x^*, namely: constraints 1 and 2; constraints 1 and 3; constraints 2 and 3. However, note that not every choice of defining constraints for x^* provides an optimality proof. Indeed, consider the three cases.

- Constraints 1 and 2. In this case, to prove optimality we need to find multipliers y_1, y_2 such that

$$\begin{array}{rcl} y_1 \cdot (x_1 + x_2 & \leq & 1) \\ y_2 \cdot (\ x_1 \qquad & \leq & 1) \\ \hline x_1 - x_2 & \leq & 1 \end{array}$$

  Clearly the only solution is $y_1 = -1$, $y_2 = 2$. Since y_2 is negative, this choice of defining constraints does not prove optimality of x^*. Note that with this choice of defining constraints the effective constraints are 1 and 2 and there are no non-basic variables. The basic variables are x_1 and x_2 but basic variable x_2 takes value 0.

- Constraints 1 and 3. In this case, to prove optimality we need to find multipliers y_1, y_3 such that

$$\begin{array}{rrcl} y_1 \cdot (x_1 + x_2 & \leq & 1) \\ y_3 \cdot (\quad -x_2 & \leq & 0) \\ \hline x_1 - x_2 & \leq & 1 \end{array}$$

Clearly the only solution is $y_1 = 1$, $y_3 = 2$. Since y_1 and y_3 are non-negative, this choice of defining constraints proves optimality of x^*. Note that with this choice of defining constraints the effective constraint is 1 the non-basic variable is x_2. The basic variable is x_1 and it takes a positive value.

- Constraints 2 and 3. In this case, to prove optimality we need to find multipliers y_2, y_3 such that

$$\begin{array}{rrcl} y_2 \cdot (x_1 \quad & \leq & 1) \\ y_3 \cdot (\quad -x_2 & \leq & 0) \\ \hline x_1 - x_2 & \leq & 1 \end{array}$$

Clearly the only solution is $y_2 = 1$, $y_3 = 1$. Since y_2 and y_3 are non-negative, this choice of defining constraints proves optimality of x^*. Note that with this choice of defining constraints the effective constraint is 2 and the non-basic variable is x_2. The basic variable is x_1 and it takes a positive value.

Figure 1.5 illustrates the three LPs obtained by only considering each of the three possible choices of the defining constraints. As can be seen from the figure, for the LP comprised of only constraints 1 and 2 the point $x^* = (0, 1)$ is not optimal, which explains why this choice of defining constraints does not prove optimality. For the LP comprised of only constraints 1 and 3, instead, the point $x^* = (1, 0)$ is optimal, and similarly for the LP comprised of only constraints 2 and 3.

Constraint 3, nonnegativity constraint $x_2 \geq 0$ needs to be in the defining set of constraints for point x^* otherwise we cannot prove optimality and we end up with a basic variable x_2 taking 0 value.

1.5 Multiple optimal solutions

It may happen that an objective function reaches its optimum value at more than one extreme point. Consider the LP

$$\begin{array}{rrcl} \text{maximise} & 3x_1 + 3x_2 & & \\ \text{subject to} & x_1 + x_2 & \leq & 5 \\ & 2x_1 - x_2 & \leq & 5 \\ & x_2 & \leq & 3 \\ & x_1, x_2 & \geq & 0 \end{array}$$

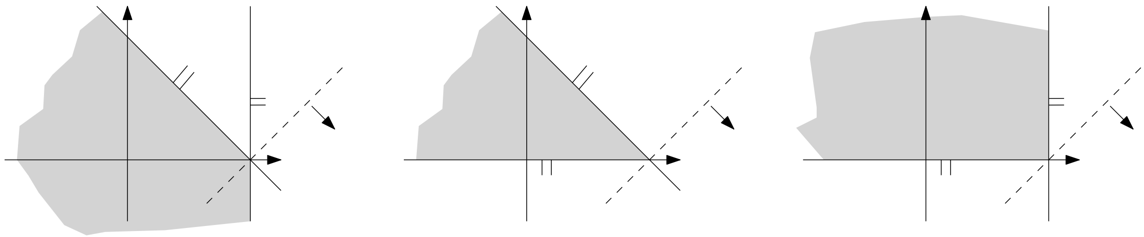


Figure 1.5: From left to right, the above diagrams represent the LPs comprised only of, respectively, constraints 1 and 2, constraints 1 and 3, constraints 2 and 3.

which is illustrated in the diagram in Figure 1.6. The optimal extreme solutions are points P^*, P^{**}, and all points in the line segment between P^* and P^{**} are also optimal, but not extreme. Any computer code will find only one of these points. The question of interest is: if we have the solution at one point can we tell whether or not another optimal point exists?

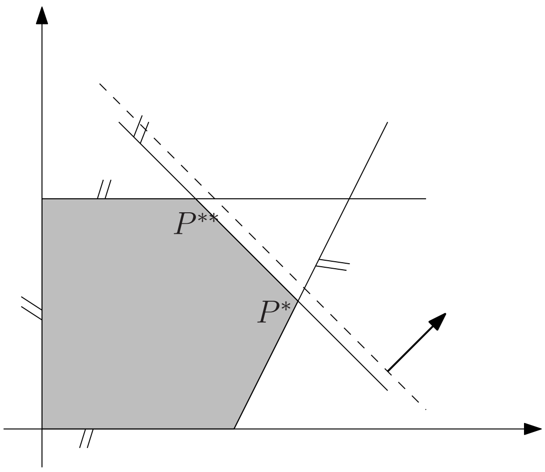


Figure 1.6: A problem with multiple optimal solutions: all points in the line segment between P^* and P^{**} are optimal.

At point P^* the point is defined by constraints 1 and 2. Solving the equations to find the intersection point gives point P^* as $(\frac{10}{3}, \frac{5}{3})$, and a value of the objective function of 15. To determine the value of the dual variables to prove optimality of the extreme point P^*, we need to solve the system

$$\begin{aligned} y_1 + 2y_2 &= 3 \\ y_1 - y_2 &= 3 \end{aligned}$$

which gives $y_1 = 3$, $y_2 = 0$. The dual values of the constraints not defining point P^* are 0. Note that one of the dual value of defining constraint 2 is zero. This is because constraint 1 is parallel to the objective contours and thus it is the only one needed to generate the objective function coefficients.

At point P^{**} the point is defined by constraints 1 and 3. Solving the equations to find the intersection point gives point P^{**} as $(2, 3)$, and a value of the objective function of 15. To determine the value of the dual variables to prove optimality of the extreme point P^{**}, we need to solve the system

$$\begin{aligned} y_1 \quad &= 3 \\ y_1 + y_3 &= 3 \end{aligned}$$

which gives $y_1 = 3$, $y_3 = 0$. The dual values of the constraints not defining point P^{**} are 0. Again note that only constraint 1, which is parallel to the objective, has positive dual value.

Note that, at either point, the dual value of one of the constraints is 0. Indeed, the only constraint with a positive dual value is constrain 1, which defines the line where both P^* and P^{**} lie. In general, in order for the problem to have multiple optimal solutions, the following condition must be met.

Necessary condition for multiple optimal solutions
Given an LP problem, let x^* be an optimal extreme solution. If the problem has multiple optimal solutions then some constraint defining x^* must have dual value 0.

In general, the above condition is necessary but not sufficient for multiple optimal solutions to exist. This means that there are examples where some of the constraints defining an extreme optimal solutions have zero dual value, yet there is only one optimum. The following example illustrates this situation.

$$\begin{aligned} \text{maximise} \quad & 3x_1 + 3x_2 \\ \text{subject to} \quad x_1 + x_2 &\leq 5 \\ 2x_1 - x_2 &\leq 5 \\ x_2 &\leq 3 \\ -x_1 + 2x_2 &\leq 0 \\ x_1, x_2 &\geq 0 \end{aligned}$$

From Figure 1.7, it appears that the unique optimal solution is the point Q, which lies at the intersection of constraints 1, 2, and 4. To prove optimality of point Q, we need to choose two independent constraints among 1, 2 and 4 as defining constraints.

If we select constraints 1 and 2 as the defining ones, the corresponding dual values are $y_1 = 3$, $y_2 = 0$ (the dual values of all remaining constraints being zero). This proves that point Q is optimal, because $y_1, y_2 \geq 0$, but it would not allow us to conclude that Q is the unique optimum, since constraint 2 defines Q but it has a dual value of 0.

Note that, if instead we selected constraints 2 and 4 as defining constraints, the corresponding dual values would be $y_2 = 3$, $y_4 = 3$ (the dual values of all remaining constraints being zero). This proves that the point Q is optimal, because $y_2, y_3 \geq 0$, but also that there is no other optimal solution, because $y_2, y_3 \neq 0$.

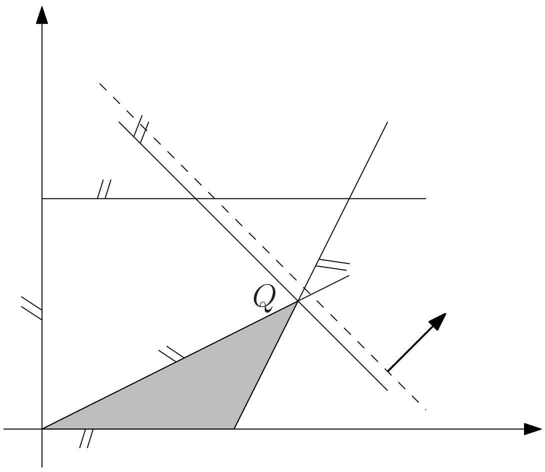


Figure 1.7: A problem with a unique optimum but a dual solution with dual value 0 on some constraint defining the optimal solution.

Note that when we selected constraints 1 and 2, we ignored constraint 4. Without constraint 4 Q is still an optimal solution but it is not a unique optimum. However when we selected constraints 2 and 4, we ignored constraint 1, and without constraint 1 Q is not only an optimal solution but it is the unique optimum.

Chapter 2

The simplex method

Linear programming emerged after the end of world war II as a practical, powerful tool in a wide array of applications. This was made possible by the convergence of two events: the advent of computers, and the development of the first effective method for solving LP problems, the Simplex Method. Devised by George Dantzig in the 1940's, the method remains today at the core of all commercial Linear Programming solvers. Solvers such as IBM's Cplex, Gurobi, or FICO's Xpress, to name a few, all implement some version of the method.

2.1 The Simplex Method: example

Consider the following LP problem

$$\begin{array}{rrcrcr} \max & 3x_1 & + & 2x_2 & & \\ s.t. & x_1 & + & x_2 & \leq & 6 \\ & x_1 & + & 2x_2 & \leq & 10 \\ & x_1 & - & x_2 & \leq & 4 \\ & & & x_1, x_2 & \geq & 0 \end{array} \tag{2.1}$$

Let us first transform the problem in standard equality form:

$$\begin{array}{rrcrcrcrcrcr} \max & 3x_1 & + & 2x_2 & & & & & & & & \\ s.t. & x_1 & + & x_2 & + & x_3 & & & & & = & 6 \\ & x_1 & + & 2x_2 & & & + & x_4 & & & = & 10 \\ & x_1 & - & x_2 & & & & & + & x_5 & = & 4 \\ & & & & & & & & x_1, x_2, x_3, x_4, x_5 & \geq & 0 & \end{array} \tag{2.2}$$

Note that there is an obvious feasible solution to start from, namely the point $x^* = (0, 0, 6, 10, 4)$ with objective function of 0. The solution x^* is extreme in $\mathbb{R}^5$ since all three resource constraints and the two non-negativity constraints for the variables x_1, x_2 in the objective are defining for x^* (they are independent and satisfied at equality). The equivalent solution $(0, 0) \in \mathbb{R}^2$ in the original problem (2.1) is also extreme.

Note that in (2.2) the non-basic variables x_1, x_2 are in the objective and the basic variables x_3, x_4, x_5 are written is terms of the non-basic ones, where the coefficient of the basic variables in the resource constraints is +1. This way the basic solution corresponding to (2.2) is given by setting to zero the non-basic variables, which gives the basic variables the value of the right hand side parameters. This layout of the LP is called a *dictionary*. And the obvious solution x that arises by setting the variables in the objective to 0 is called the *dictionary solution*.

If the original problem is in standard form with n variables and m resource constraints then in standard equality form it will have $n + m$ variables and m equality resource constraints. The dictionary solution x satisfies m resource constraints and n nonnegativity constraints at equality; let us assume that these are independent then the dictionary solution x is a basic point. We define a *basis* to be the set of the basic variables of the dictionary solution. Thus any variables not in the basis have to be set to 0 (the non-basic variables). If the problem is not degenerate then there is a one to one correspondence between a basis and a dictionary/ dictionary solution. The basis that corresponds to (2.2) is $B = \{3, 4, 5\}$. When the solution that corresponds to a basis or a dictionary is feasible then we call it a *feasible basis* and a *feasible dictionary*.

First iteration Going back to the example (2.2), can we find a better solution?

Note that, if we increase x_2 by $t \geq 0$ and leave $x_1 = 0$, the objective value increases by $2t$.

However, to insure that the resource constraints are satisfied, the remaining components must become

$$\begin{aligned} x_3(t) &= 6 - t \\ x_4(t) &= 10 - 2t \\ x_5(t) &= 4 + t \end{aligned}$$

We can only increase t as long as the above three variables take nonnegative values. What is the maximum value of t we can choose? In order to have $x_3(t) \geq 0$, we need $t \leq 6$. For $x_4(t) \geq 0$, we need $t \leq 5$. Note instead that $x_5(t) \geq 0$ for all nonnegative values of t. Thus the largest t we can choose is $t = 5$.

The new solution we obtain is therefore the point $(0, 5, 1, 0, 9)$. Note that there are still three effective constraints and two non-basic variables (namely, x_1, x_4). The solution is extreme. The value of the solution is $2 \cdot 5 = 10$. Here we say that x_2 is the entering variable, i.e. it enters the basis and from non-basic it becomes basic; and x_4 is the leaving variable, i.e. it leaves the basis and from basic it becomes non-basic. The new basis is $B = \{2, 3, 5\}$.

Second iteration Can we find a yet better solution? What we did before, was to increase the value of a nonbasic variable with positive coefficient in the objective function.

In (2.2) the objective and the basic variables were expressed in terms of the non-basic variables. So setting the non-basic variables to zero leaves the basic variables to take the value of the right hand side, where if positive then the solution is feasible. So when we increased the value of x_2, the basic variables x_3, x_4, x_5 were adjusted in value to maintain feasibility. But their change in value did not affect the objective since they do not appear in the objective. This guarantees that by increasing a non-basic variable with positive objective coefficient the new solution value we find will have better objective value.

So we would like only the nonbasic variables x_1, x_4 to appear in the objective function. To accomplish this, we resort to the following trick. Let us introduce a new variable z, and set

$$z = 3x_1 + 2x_2.$$

Now, the original problem can be stated as

$$\begin{array}{rrcrcrcrcrcl}
\max z \\
z = & 3x_1 & + & 2x_2 \\
 & x_1 & + & x_2 & + & x_3 & & & & & = & 6 \\
 & x_1 & + & 2x_2 & & & + & x_4 & & & = & 10 \\
 & x_1 & - & x_2 & & & & & + & x_5 & = & 4 \\
 & & & & & & & & & x_1, x_2, x_3, x_4, x_5 & \geq & 0
\end{array}$$

Note that the second constraint of the original problem expresses x_4 in terms of the original non-basic variables x_1, x_2 and thus we can solve for the new basic variable x_2 in terms of the new nonbasic variables x_1, x_4, that is

$$x_2 = 5 - \frac{1}{2}x_1 - \frac{1}{2}x_4.$$

Substituting x_2 with $5 - \frac{1}{2}x_1 - \frac{1}{2}x_4$ in the equation defining z, we obtain

$$z = 10 + 2x_1 - x_4.$$

Therefore, if we increase the value of x_1 from 0 to t, while leaving the value of x_4 at 0, the next solution will have value $10 + 2t$, which is greater than the current one. We need to compute the values of the basic variables in order to insure that the next solution satisfies the resource constraints. In order to do this, we need to know how x_2, x_3, x_5 change as x_1 increases. Thus, we need to solve the basic variables in terms of the nonbasic one. This can be done by substituting x_2 with $5 - \frac{1}{2}x_1 - \frac{1}{2}x_4$ in all the constraints of the problem.

We obtain the following problem, equivalent to the previous one.

$$
\begin{array}{rrcrcrcrcrcl}
\max z & & & & & & & & & & & \\
z = 10+ & 2x_1 & & & & & - & x_4 & & & & \\
 & \frac{1}{2}x_1 & & & + & x_3 & - & \frac{1}{2}x_4 & & & = & 1 \\
 & \frac{1}{2}x_1 & + & x_2 & & & + & \frac{1}{2}x_4 & & & = & 5 \\
 & \frac{3}{2}x_1 & & & & & + & \frac{1}{2}x_4 & + & x_5 & = & 9 \\
 & & & & & & & x_1, x_2, x_3, x_4, x_5 & & & \geq & 0
\end{array}
\tag{2.3}
$$

Note that in (2.3) the basic variables x_2, x_3, x_4 are written with respect to the non-basic variables and the thus values of the basic variables are given by the right hand side values. This is the dictionary that corresponds to the dictionary solution $(0, 5, 1, 0, 9)$ and to the basis $B = \{2, 3, 5\}$.

If we increase x_1 by $t \geq 0$ and leave $x_4 = 0$, the objective value increases by $2t$. The remaining components must become

$$
\begin{aligned}
x_2(t) &= 5 - \frac{1}{2}t \\
x_3(t) &= 1 - \frac{1}{2}t \\
x_5(t) &= 9 - \frac{3}{2}t
\end{aligned}
$$

The variables x_2, x_3, x_5 remain nonnegative as long as t satisfies, respectively, $t \leq 10$, $t \leq 2$, and $t \leq 6$. Thus the largest t for which the solution is nonnegative is $t = 2$. The new solution is $(2, 4, 0, 0, 6)$, with value 14. The variables x_1, x_2, x_5 are basic, while x_3, x_4 are nonbasic. In this iteration x_1 is the entering basic variable, while the variable x_3 is the leaving basic variable. The new basis is $B = \{1, 2, 5\}$.

Third iteration As before, we want to express z and the basic variables x_1, x_2, x_5 in terms of the nonbasic variables x_3, x_4. We can use the constraint $\frac{1}{2}x_1 + x_3 - \frac{1}{2}x_4 = 1$ to express x_1 in terms of x_3 and x_4, namely

$$x_1 = 2 - 2x_3 + x_4.$$

Substituting x_1 with $2 - 2x_3 + x_4$ in all constraints of the previous LP, we get

$$
\begin{array}{rrcrcrcrcrcl}
\max z = 14 & & & & - & 4x_3 & + & x_4 & & & & \\
 & x_1 & & & + & 2x_3 & - & x_4 & & & = & 2 \\
 & & + & x_2 & - & x_3 & + & x_4 & & & = & 4 \\
 & & & & - & 3x_3 & + & 2x_4 & + & x_5 & = & 6 \\
 & & & & & & & x_1, x_2, x_3, x_4, x_5 & & & \geq & 0
\end{array}
$$

The above dictionary corresponds to the dictionary solution $(2,4,0,0,6)$ and basis $B = \{1,2,5\}$.

If we increase x_4 by $t \geq 0$ and leave $x_3 = 0$, the objective value increases by t. The remaining components must become

$$\begin{aligned} x_1(t) &= 2 + t \\ x_2(t) &= 4 - t \\ x_5(t) &= 6 - 2t \end{aligned}$$

The largest value of t for which the basic variables are nonnegative is $t = 3$. The new basic solution is $(5,1,0,3,0)$, with value 17 and the new basis is $B = \{1,2,4\}$.

Fourth iteration Substituting x_4 with $3+\frac{3}{2}x_3-\frac{1}{2}x_5$ in all constraints, we obtain

$$\begin{array}{llllllllll} \max z = 17 & & & & - & \frac{5}{2}x_3 & & & - & \frac{1}{2}x_5 & & \\ & x_1 & & & + & \frac{1}{2}x_3 & & & + & \frac{1}{2}x_5 & = & 5 \\ & & + & x_2 & + & \frac{1}{2}x_3 & & & - & \frac{1}{2}x_5 & = & 1 \\ & & & & - & \frac{3}{2}x_3 & + & x_4 & + & \frac{1}{2}x_5 & = & 3 \\ & & & & & & & & & x_1, x_2, x_3, x_4, x_5 & \geq & 0 \end{array}$$

The above dictionary corresponds to the dictionary solution $(5,1,0,3,0)$ and the basis $B = \{1,2,4\}$.

Can there be a better solution? Since the coefficient of x_3 and x_5 in the objective function are negative, it is impossible to achieve a value better than 17. Therefore our current solution is optimal.

We have visited the following solutions:

$$(0,0,6,10,4) \to (0,5,1,0,9) \to (2,4,0,0,6) \to (5,1,0,3,0).$$

The components relative to the original variables x_1 and x_2 can be plotted in a diagram, shown in Figure 2.1.

2.2 The simplex method

In order to be applied, the simplex method requires the problem to be converted initially into standard equality form. Note that LP solvers accept LP problems in any form, so the user needs not be concerned with entering the problem in the appropriate form.

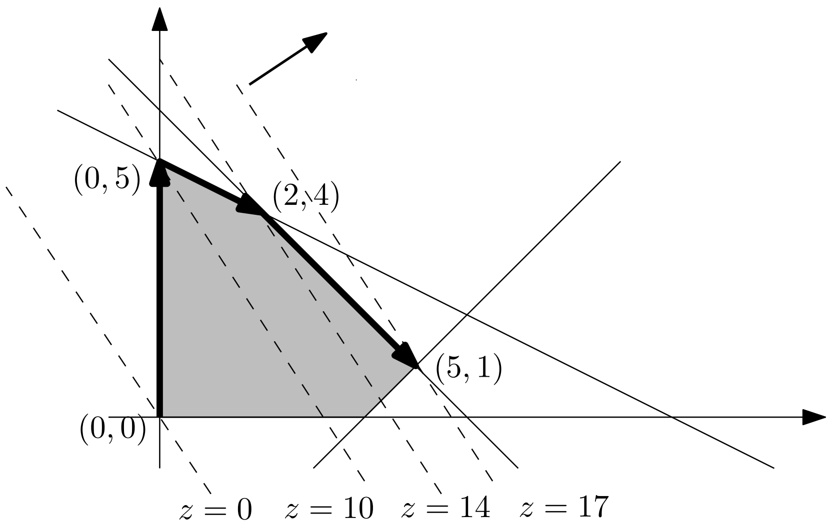


Figure 2.1: Sequence of points visited by the simplex.

Thus, we assume that we are dealing with an LP problem in the form

$$\begin{aligned} \max cx & \\ Ax = &\ b \\ x \geq &\ 0 \end{aligned} \tag{2.4}$$

where A is an $m \times (n+m)$ matrix, c is an $(n+m)$-dimensional row vector, b is an m-dimensional column vector, and x is a column vector of $n+m$ variables. We will also assume that the m resource constraints are independent. Since all resource constraints are equalities and they are independent, every basic solution will have m effective constraints and thus m basic variables and n nonbasic variables.

It will be convenient to write (2.4) in the following equivalent form:

$$\begin{aligned} \max z & \\ z = cx & \\ Ax = &\ b \\ x \geq &\ 0 \end{aligned} \tag{2.5}$$

The Simplex Method needs to be given an initial extreme solution to start form, say $\bar{x}$, and the corresponding set of basic variables (in the example in the previous section the initial extreme solution was the point $(0, 0, 6, 10, 4)$).

Let us assume that the basic variables are $x_{B[1]}, x_{B[2]} \ldots, x_{B[m]}$, belonging to basic B and let us denote by N the set of indices of the n nonbasic variables. In the previous example, for the extreme point $(0, 0, 6, 10, 4)$, $x_{B[1]} = x_3$, $x_{B[2]} = x_4$, $x_{B[3]} = x_5$ corresponding to basis $B = \{3, 4, 5\}$, and $N = \{1, 2\}$.

We solve the variables z, $x_{B[1]}, x_{B[2]} \ldots, x_{B[m]}$ in terms of the nonbasic variables, so that the problem is in dictionary form:

$$
\begin{array}{llllll}
\max & z & & & & \\
& z = & \bar{z} & + \sum_{j \in N} \bar{c}_j x_j & & \\
& & x_{B[i]} & + \sum_{j \in N} \bar{a}_{ij} x_j & = & \bar{b}_i, \quad i = 1, \ldots, m \\
& & x & & \geq & 0.
\end{array} \tag{2.6}
$$

The extreme solution or dictionary solution corresponding to the above dictionary $\bar{x}$ is defined by

$$
\begin{array}{rcl}
\bar{x}_{B[i]} & = & \bar{b}_i; \quad i = 1, \ldots, m \\
\bar{x}_j & = & 0; \quad j \in N
\end{array}
$$

(so in particular $\bar{b}_1, \ldots, \bar{b}_m \geq 0$), and has objective value $\bar{z}$. The values $\bar{c}_j$ of the

dictionary that corresponds to basis B are called *reduced costs* corresponding to basis B for the non-basic variables in N. The reduced costs corresponding to basis B for the basic variables are set to 0. Reduced costs have a nice intuitive interpretation: it is the marginal increase in the objective value when one of the non-basic variables (now set to 0) is increased.

There are two possible cases:

Case 1. There exists a nonbasic variable x_k such that $\bar{c}_k > 0$ (reduced cost is positive).

Ideally, we would like to find a new feasible basis whose objective value is better (or not worse) than $\bar{z}$. Note that, since $\bar{c}_k > 0$, if we increase the value of x_k from 0 to some number $t \geq 0$, while leaving at 0 the value of all other nonbasic variables, then the value of the new solution in the objective function will increase by $\bar{c}_j t$, thus improving the value of the objective function. However, in order to do so, we need to adjust the values of the basic variables to maintain feasibility.

More formally, for $t \geq 0$, the basic variables need to be defined as

$$
x_{B[i]}(t) \;=\; \bar{b}_i - t\bar{a}_{ik}, \qquad i = 1, \ldots, m; \tag{2.7}
$$

By construction, the new solution satisfies the resource constraints. We would like to increase the value of t as much as possible. The only thing preventing t from increasing indefinitely is the nonnegativity of the variables. Thus we can have two sub-cases

a) For some value of $t = t^*$, the value of some basic variable, say x_ℓ, goes to 0, while keeping the values of all other variables nonnegative. In this

case, the new extreme solution is obtained by replacing t^* into (2.7). The variable x_k becomes basic, while the variable x_ℓ become nonbasic, and we repeat.
(In the first iteration in the example in the previous section, $x_k = x_2$, $x_\ell = x_4$, $t^* = 5$, and the new solution is $(0, t^*, 6 - t^*, 0, 4 + t^*)$.)

b) The values of the variables remain nonnegative for any positive value of t.
In this case we can increase the value of t as much as we want, thus obtaining feasible solutions of arbitrarily large value. It follows that the problem is unbounded.

Case 2. $\bar{c}_j \leq 0$ for all indices $j \in N$.
This means that all coefficient in the objective function of the LP (2.6) are ≤ 0. So no solution can have value greater than $\bar{z}$. Since $\bar{x}$ has value $\bar{z}$, it must be optimal. Thus the algorithm stops.

The Simplex Method

Start from an extreme solution, with basic variables $x_{B[1]}, \ldots, x_{B[m]}$.

1. Write the LP in dictionary form (2.6) by expressing the variables $z, x_{B[1]}, \ldots, x_{B[m]}$ in terms of the nonbasic variables.
2. If $\bar{c}_j \leq 0$ for all $j \in N$, then the current solution is optimal, STOP.
3. Otherwise, pick a nonbasic variable x_k such that $\bar{c}_k > 0$.
Compute the largest value t^* of t such that the solution (2.7) is feasible.
   3a. If $t^* < +\infty$, then some basic variable x_ℓ takes value 0. Compute the new extreme solution, and replace x_ℓ with x_k as basic variable. Return to 1.
   3b. If $t^* = +\infty$, then the problem is unbounded. STOP.

2.3 Issues that arise when using the simplex method

In this section we discuss problems that one might face when running the simplex method. For example, how do we pick an entering basic variable? What happens when the problem is degenerate? What if the problem is unbounded? What happens if there are multiple optima?

1. **Tie in the nonbasic entering variables** We did not specify how to choose which nonbasic variable to increase. In example (2.1), in the first iteration of the simplex algorithm we chose non-basic variable x_2 to become a basic variable, which had coefficient 2, but we could have also chosen non-basic variable x_1, which had a higher coefficient of value 3.

   If we chose the non-basic variable with the highest coefficient this will have the highest marginal increase in the objective value. However, this does not mean that it will result to the highest total increase in the objective since we might not be able to increase that nonbasic variable by a large amount whereas another basic variable with a smaller objective coefficient might result in a higher total increase in the objective.

2. **Degeneracy** When a problem is degenerate this might result in a basic variable taking the value of zero (a degenerate variable). This could be problematic because of the following. Suppose that one of these degenerate basic variables retains its value of zero until it is chosen at a subsequent iteration to be a leaving basic variable. But the corresponding entering basic variable cannot be increased in value without the leaving basic variable becoming negative. In this case the basis will change but the value of the objective will not change because we were unable to increase the value of the entering basic variable. If z remains the same in an iteration there is a chance that the simplex method may go around in a loop repeating the same sequence of solutions periodically without increasing the objective value.

   Fortunately, though a perpetual loop is possible in theory, it rarely occurs in practice. Further, one could always change the choice of the leaving basic variable to get out of the loop. Finally, special rules have been constructed for breaking ties of leaving basic variables so that such loops are always avoided.

   For an example of this please see exercises 3 and 4 and corresponding solutions.

3. **Multiple Optima** Some LPs have multiple optimal extreme solutions. When this happens the optimal dictionary will have a nonbasic variable with 0 coefficient. To find the other extreme optimal solutions you can increase the value of this nonbasic variable. See exercise 5 for an example.

MA423 - Fundamentals of Operations Research
Lecture 4: Integer programming

Katerina Papadaki[1]

Academic year 2017/18

[1]London School of Economics and Political Science. Houghton Street London WC2A 2AE (k.p.papadaki@lse.ac.uk)

Contents

Chapter 1

Integer programming: basic concepts and solution methods

1.1 Introduction

A strong limitation in the applicability of linear programming is that the variables can take arbitrarily fractional values in an optimal solution. In most practical situations however, many quantities are only allowed to take discrete values. For example, it usually does not make sense to return a fractional answer, if a company has to decide about the number of storage facilities to open, or about the number of vehicles to purchase.

Such problems can be formulated as *integer programs (IPs).* The problem has several variants. By a *pure integer program*, we mean a linear program with all variables restricted to be integer:

$$\begin{aligned} \max\ & cx \\ \text{s. t. }\ & Ax \le b \\ & x \in \mathbb{Z}^n \end{aligned}$$

In practice, we usually have a mixture of continuous and discrete variables. This is called a *mixed integer program.* There are n_1 continuous and n_2 integer variables; the vector of variables, the objective vector and the matrix are partitioned in two parts accordingly: $x = (x_1, x_2)$, $c = (c_1, c_2)$, $A = (A_1, A_2)$.

$$\begin{aligned} \max\ & c_1x_1 + c_2x_2 \\ \text{s. t. }\ & A_1x_1 + A_2x_2 \le b \\ & x_1 \in \mathbb{R}^{n_1}, x_2 \in \mathbb{Z}^{n_2} \end{aligned}$$

As an important special case, we can have *binary variables* with only two possible values, 0 and 1. These are also called *indicator variables* as they can model *yes-no* decisions, indicating the choice of certain resources. E.g. $x_s = 1$ expresses the fact that we open a certain storage facility s, and $x_s = 0$ means that we do not open it.

There is an important tradeoff between linear programming (LP) and integer programming (IP). LP is less powerful for modelling as we cannot enforce discrete values in a solution, that can be done in IPs. On the other hand, there are highly efficient algorithms for LP (such as the Simplex algorithm). There are several algorithms that are not only efficient in practice, but also have theoretical guarantees that they terminate rapidly in terms of the input problem size. In contrast, there are no such guarantees at all for integer programming (it is a 'hard' problem in a rigorous mathematical sense of hardness). Whereas several algorithms perform reasonably well in practice, they are slower than linear programming algorithms often by orders of magnitudes, and moreover, we do not have theoretical guarantees that they terminate in a reasonable amount of time.

The most fundamental techniques for solving integer programs are Branch-and-Bound and the Cutting Plane Method; we give an overview of the Branch-and-Bound method in this lecture. Integer programming is implemented in many commercial solvers, that are able to solve typical problems of remarkable size efficiently (but much slower than LPs). Linear programming can be thought of as a swiss army knife due to its simplicity and efficiency. While formulating models is also simple for integer programming, the solution methods are far from simple. They are built on decades of extensive research and expertise. Rather than a swiss army knife, an IP solver could be compared to a contractor with a large toolbox and years of practical experience - yet we might still not be completely sure that he is able to repair our household problem. Moreover, his work is much more expensive than to fix it ourselves.

For this reason, we should always try to use as few integer variables as necessary: use them only if it is really needed and try to keep their number and usage limited.

1.2 Solving Integer Programs

In this section, we describe the fundamental Branch and Bound method for solving integer programs.

Let us start with a simple example.

$$\begin{aligned} \max\ & 2x_1 + 5x_2 \\ & 12x_1 + 5x_2 \le 60 \\ & 2x_1 + 10x_2 \le 35 \\ & x_1, x_2 \ge 0 \quad \text{and integer} \end{aligned}$$

The feasible region is illustrated in Figure 1.1 below, where the dots are the feasible integer points. The linear program optimum is at the point $x_1 = 3.864$, $x_2 = 2.727$ and the objective function $2x_1 + 5x_2$ is 21.364. Visually we can see that the possible candidate optimum points are $(2, 3)$ with objective value 19, and $(4, 2)$ with objective value 18 - hence the optimal solution is $x_1 = 2$, $x_2 = 3$.

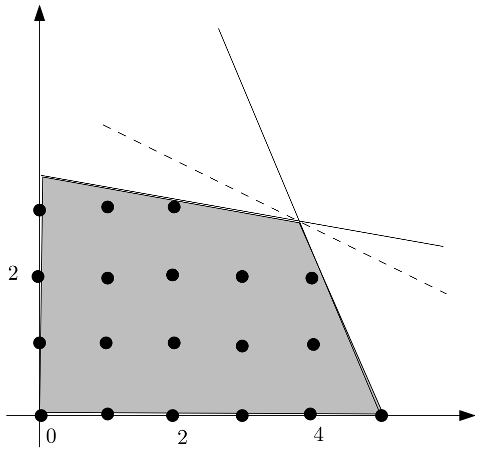


Figure 1.1

For LPs, the solution methods were based on the fact that every local optimal solution is also a global optimum. This is also not true for integer programs. For example, the point $(4, 2)$ does not have any integer neighbours that would be feasible and also give a better objective value - yet it is not optimal. In the LP case we used dual values to verify optimality but here we cannot do that. There is no simple certificate to convince yourself that you have found an optimal solution.

1.2.1 Linear programming relaxation

The difference between an LP problem and an IP problem consists of the fact that the IP has additional constraints, namely the integrality constraints. A problem B that is derived from a problem A by not taking into account some of the integrality constraints is called a *relaxation* of this problem. When we disregard the integrality constraints of an IP problem, we arrive at a normal LP problem. Therefore, such an LP problem is called the LP relaxation of the original IP problem. As the LP relaxation has a larger feasible region than the original IP, the objective function value of an IP can never be better than that of the LP relaxation of the problem. In other words: For a maximization (minimization) problem, the LP relaxation of an IP problem provides an upper (lower) bound for the objective function value of the optimal solution of the IP problem. On Figure 1.1, the gray region is the LP relaxation, and 21.364 is the optimal value of the relaxed problem, which is an upper bound to the integer optimal value of 19.

Given the fractional optimal solution, we might try rounding it to the closest integer solution. However, this might be infeasible, or feasible but not optimal. In the above example, the fractional optimal solution was $(3.864, 2.727)$. Rounding it would give $(4, 3)$, that is infeasible as in the second constraint we get $38 \leq 35$. Even if we round it down to get $(3, 2)$, which is feasible, the solution has objective value 16, which is far from 19. So even if the rounded solution is feasible, it might be far from optimal.

Yet sometimes we might get an acceptable solution by rounding. This is possible if

- The rounded solution is feasible, *and*

- The difference between the optimal objective value of the LP relaxation and the objective function value of the rounded feasible solution is acceptable for the practical purpose that the model serves.

If the variables are binary then $(\frac{1}{3}, \frac{1}{2})$ could round to $(0, 1)$ (if feasible) and the difference in objective values could be huge. Also, when rounding binary variables in many dimensions then rounding to a feasible solution could prove to be really hard.

1.2.2 The Branch and Bound algorithm

A lot of research has been and is being applied to methods to solve integer programming problems. The most successful approach to date the is *branch and bound*, developed in 1960 by Ailsa Land and Alison Doig, at the LSE Operational Research Department (the predecessor of the Operations Research Group). All the commercial mathematical programming systems offer an integer programming solution procedure based on some variation of this method.

Branch and bound can be thought of as a systematic exploration of the feasible region and an elimination of those parts which can be shown either not to contain an integer solution or not to contain the optimal solution.

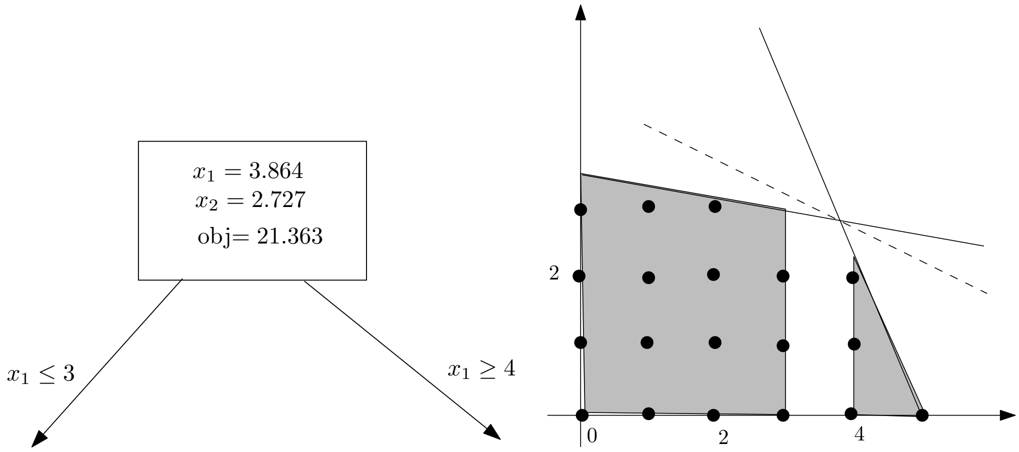
The branch and bound algorithm is a divide-and-conquer approach: the feasible region is partitioned into a collection of smaller regions. Each region is itself represented by linear constraints and the objective function can be optimised over these smaller regions. The partitions of the feasible region can themselves be further partitioned into even smaller regions. In practice the partitioning of the feasible region is performed sequentially. The algorithm is best represented by a tree graph.

An example of the Branch and Bound algorithm

Let us return to the example we considered earlier. The algorithm starts by solving the LP relaxation. This corresponds to the *root node* of the *branch and bound tree*. If the solution satisfies the integer condition we stop (this is the optimal integer solution), otherwise it is necessary to branch. Select a variable whose value is non-integer and create two new sub-problems. The constraints of the new sub-problems are chosen so that the current non-integer solution is infeasible in both sub-problems.

In the example, the solution at the root node is: $x_1 = 3.864$, $x_2 = 2.727$ with objective value 21.363. This objective value is an upper bound on the value of the optimal integer solution x^*: $z(x^*) \leq 21.363$. In fact, since the objective coefficients of the IP are integer we know that the optimal objective value is integer and thus we can round down 21.363 to 21 and get the upper bound: $z(x^*) \leq 21$. This step is called the *bounding step.*

We arbitrarily choose to branch on x_1, since x_1 is fractional (non-integer). This is done by creating two sub-problems: one with the extra constraint $x_1 \leq 3$ and the other with $x_1 \geq 4$. (Figure 1.2(a)). The two subproblems are shown on Figure 1.2(b). This step is called the *branching step.*



(a) (b)

Figure 1.2

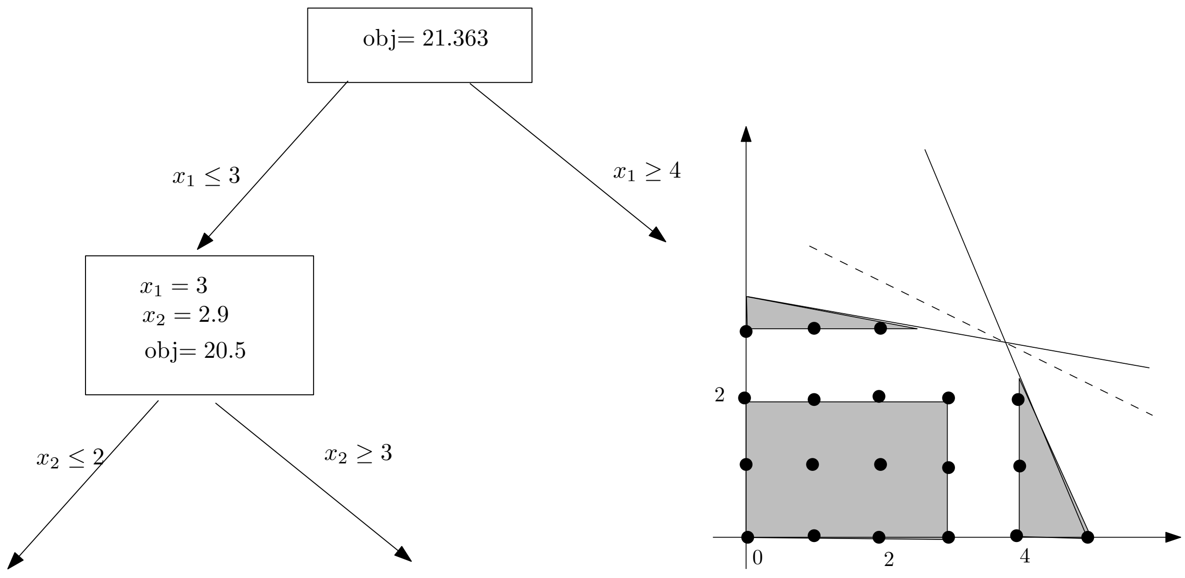
We choose (arbitrarily) to solve the left-hand problem first. This gives a solution $x_1 = 3$ and $x_2 = 2.9$ with objective value 20.5. Notice that because an additional constraint, $x_1 \leq 3$, has been added, the value of the objective function either stays the same or decreases. This gives a tighter upper bound of 20 for the integer solution of the current sub-problem (after rounding down from 20.5).

As x_2 is non-integer two further sub-problems with the constraints $x_2 \leq 2$ and $x_2 \geq 3$ are created (Figure 1.3). There are now three unsolved sub-problems.

If we now solve the leftmost problem ($x_1 \leq 3$, $x_2 \leq 2$) this gives an integer solution $x_1 = 3$, $x_2 = 2$ with objective value 16. We call this solution an *incumbent* solution, which means the best known integer solution so far. This integer solution provides a lower bound on the original problem $x^* : 16 \leq x^* \leq 21$. So now we know that the optimal IP value is an integer between 16 and 21. Note that whereas the LP relaxation gives an upper bound, finding feasible integer solutions give lower bounds. For this sub-problem we found the best integer solution so we do not branch any further. This branch or subproblem ($x_1 \leq 3$, $x_2 \leq 2$) is said to be *fathomed (conquered) by integrality.*

Now, the algorithm backtracks and selects one of the unsolved sub-problems. Arbitrarily choosing the most recently formed unsolved sub-problem ($x_1 \leq 3$, $x_2 \geq 3$)and solving it gives the tree on Figure 1.4. The objective value is 20.

Moving to sub-problem with constraints $x_1 \leq 3$, $x_2 \geq 3$, and $x_1 \leq 2$ is equivalent to solving the sub-problem with constraints $x_1 \leq 2$, $x_2 \geq 3$. This produces solution $x_1 = 2$, $x_2 = 3.1$, with objective value 19.5. Thus, the bounds for the optimal integer solutions on this subproblem are: $16 \leq x^* \leq 19$. From here we branch with respect to variable x_2 by adding constraints: $x_2 \leq 3$ and $x_2 \geq 4$. Solving the relaxation of the left subproblem with $x_2 \leq 3$, gives an integer solution $x_1 = 2$, $x_2 = 3$ with objective value 19. This is now the incumbent solution and we say that this sub-problem is *fathomed by incumbent solution.* So we do not branch on this sub-problem. Now if solve the relaxation of the right subproblem with $x_2 \geq 4$ we get an infeasible solution. Thus we will not be able to find any more solutions by branching further so we do not branch



(a) (b)

Figure 1.3

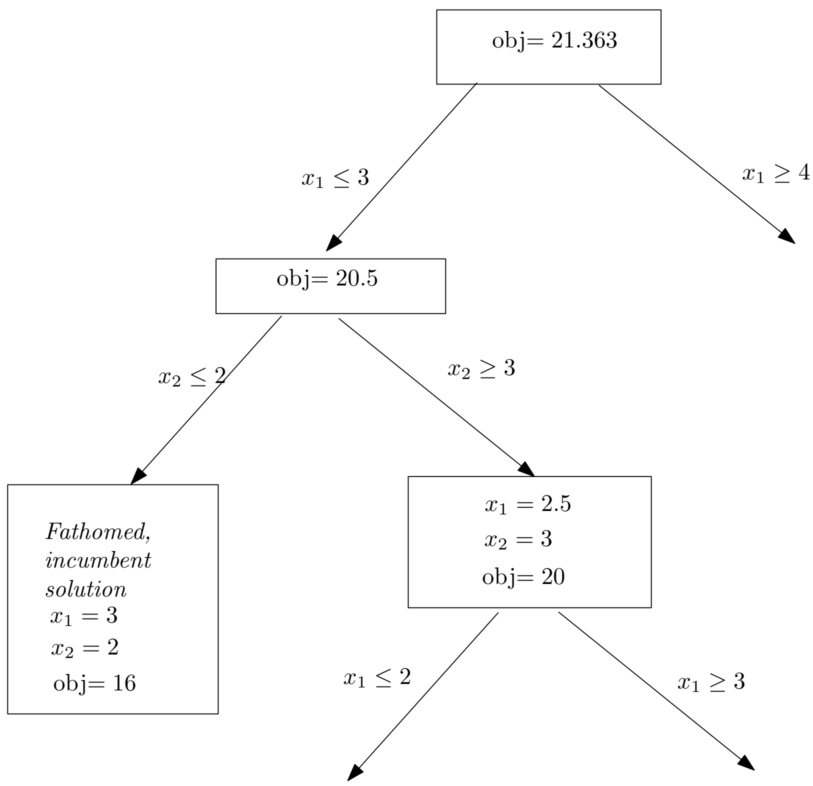


Figure 1.4

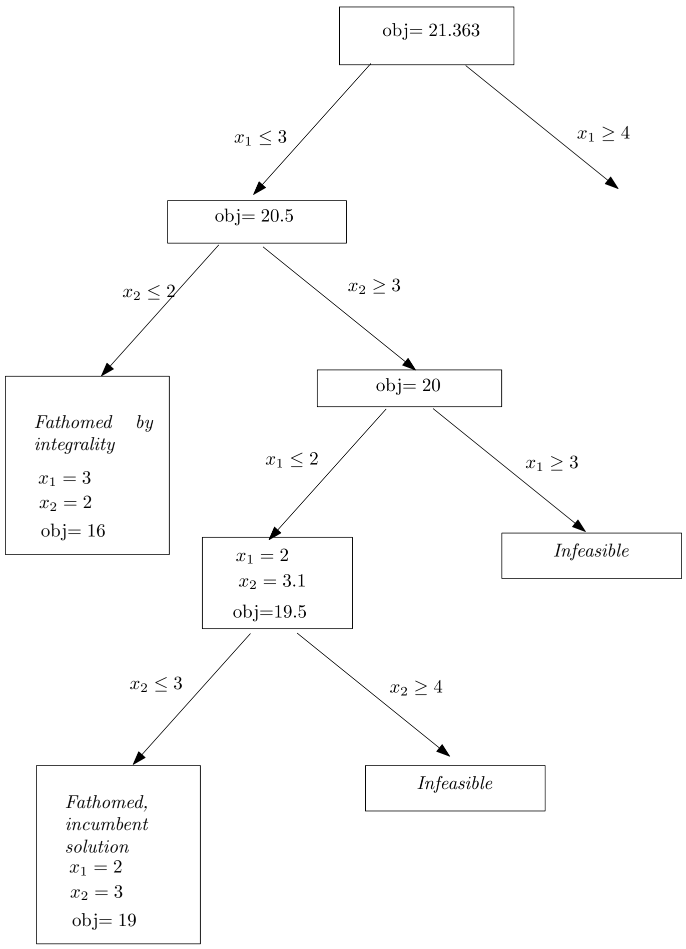

obj= 21.363
$x_1 \leq 3$
$x_1 \geq 4$
obj= 20.5
$x_2 \leq 2$
$x_2 \geq 3$
Fathomed by integrality
$x_1 = 3$
$x_2 = 2$
obj= 16
obj= 20
$x_1 \leq 2$
$x_1 \geq 3$
$x_1 = 2$
$x_2 = 3.1$
obj=19.5
Infeasible
$x_2 \leq 3$
$x_2 \geq 4$
Fathomed, incumbent solution
$x_1 = 2$
$x_2 = 3$
obj= 19
Infeasible

Figure 1.5

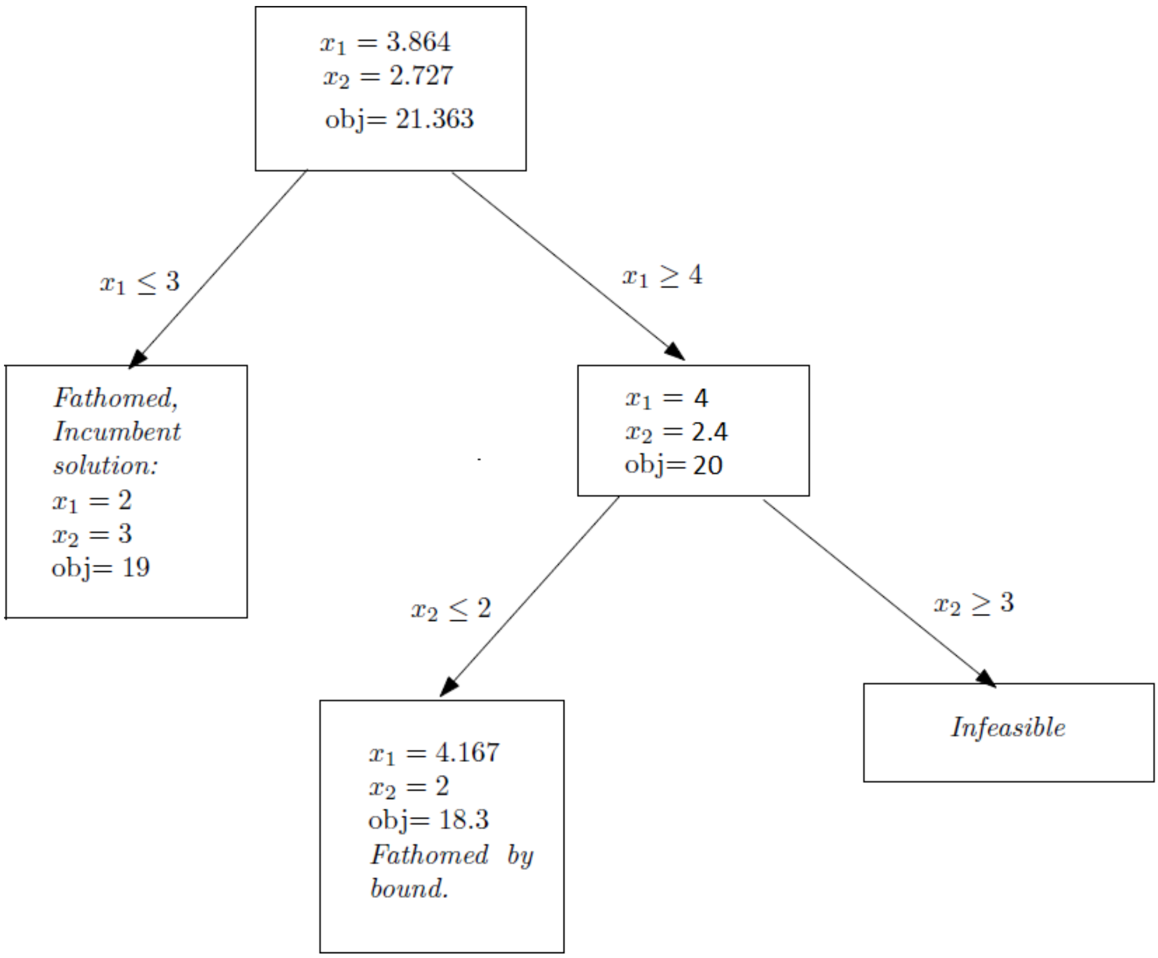

$x_1 = 3.864$
$x_2 = 2.727$
obj= 21.363
$x_1 \leq 3$
$x_1 \geq 4$
Fathomed, Incumbent solution:
$x_1 = 2$
$x_2 = 3$
obj= 19
$x_1 = 4$
$x_2 = 2.4$
obj= 20
$x_2 \leq 2$
$x_2 \geq 3$
$x_1 = 4.167$
$x_2 = 2$
obj= 18.3
Fathomed by bound.
Infeasible

Figure 1.6

on this sub-problem. We say that this branch is *fathomed by infeasibility.*

Continuing we solve the relaxation of sub-problem $x_1 \leq 3$, $x_2 \geq 3$, and $x_1 \geq 3$, which is equivalent to adding the constraints $x_1 = 3$ and $x_2 \geq 3$, and we find that the problem is infeasible. Thus we ignore this sub-problem and do not branch on it.

We have now considered all options that branch from sub-problem $x_1 \leq 3$. The resulting sub-tree is shown in Figure 1.5. The incumbent solution at this stage is $(2, 3)$ with objective value 19 and the bounds are: $19 \leq x^* \leq 21$. This means that if the optimal integer solution satisfies $x_1 \leq 3$ then it must be $(2, 3)$.

At this point, the entire branch of the tree where $x_1 \leq 3$ is completely fathomed. So we backtrack to the very beginning and solve the sub-problem with the single extra constraint $x_1 \geq 4$. Continuing in the same fashion as before and subsequently branching gives the tree on Figure 1.6. At the sub-problem with $x_1 \geq 4$, $x_2 \leq 2$, the LP optimum value is 18.3. As extra constraints are added to form the sub-problems, the value of the objective function cannot increase, it either stays the same of decreases. Consequently, in all later sub-problems on this branch we get solutions with objective value less than or equal to 18.3. However we have a better incumbent solution of value 19. Consequently, there is no point in continuing to branch. We say that the sub-problem $x_1 \geq 4$, $x_2 \leq 2$ is *fathomed by bound.* The sub-problem $x_1 \geq 4$, $x_2 \geq 3$ gives an infeasible solution, here we can say that this branch is *fathomed by infeasibility.* Thus no better solution has come out of branch $x_1 \geq 4$.

The incumbent, and thus optimal solution to the problem at the end of the entire branch and bound search is $x_1 = 2$, $x_2 = 3$ with an objective function value of 19.

In the branch and bound algorithm there are three ways that a sub-problem can be **fathomed**, which means conquered or dismissed from consideration:

1. *Fathomed by integer solution*: When the solution of the LP relaxation of the sub-problem is integer then there is no further need to branch since we have solved optimally the IP sub-problem. If this integer solution is better than the current incumbent solution then it becomes the new incumbent and we say that it is *fathomed by the incumbent solution.*

2. *Fathomed by infeasibility*: When the LP relaxation of the sub-problem has no feasible solutions then we know that the IP sub-problem is also infeasible and thus we do not branch any further.

3. *Fathomed by bound*: When the solution of the LP relaxation of the IP sub-problem has objective value (or rounded objective value if objective coefficients are integer) that is less than the objective value of the incumbent solution then we know that if we branch further we will only get worse integer solutions and thus we do not branch any further.

Branch and bound algorithm

- **Initialize**: Apply the bounding step, fathoming step and optimality test to the whole problem. If not fathomed then classify this problem as one remaining sub-problem and perform the iteration steps below:

- Iteration steps:

  1. **Branching step**: amongst the remaining unfathomed problems select one. Branch on one of the variables that did not have integer value in the LP relaxation.
  2. **Bounding step**: for each new subproblem apply the simplex algorithm to its LP relaxation to obtain an optimal solution and an objective value. If the objective coefficients are integer then round down (up for minimising) this value. This rounded objective value is an upper bound (lower bound when minimising) on the objective value of the IP sub-problem.
  3. **Fathoming step**: For each sub-problem apply the three fathoming tests summarised above and discard all the sub-problems that are fathomed by any of the test.

- **Optimality check**: Stop when there are no remaining sub-problem. The current incumbent solution is optimal.

Remarks:

1. We do not specify above how to pick the next sub-problem. You can pick the one with the highest bound (lowest when minimizing) because this sub-problem would be the most promising one to contain an optimal solution to the whole problem. Or you could pick the one that was created most recently. This is because this is the most similar to the one you just solved by simplex and simplex could use re-optimisation techniques to solve it faster.

2. If your objective coefficients are not integer then you should not round the bound in the bounding step.

3. If your variables were binary integer variables then for branching variable x_1 your branches would be equalities: $x_1 = 1$ and $x_1 = 0$.

4. Notice that in order to solve this problem in 2 variables it was necessary to solve 11 linear programs. In general, the number of steps may blow up exponentially: if we have k binary variables, we might end up with 2^k different cases. It requires substantially more work to solve an integer program than a linear program - in some cases, it takes prohibitively long. That is why great care should be taken in setting up an integer program and check whether there is possibly a way to formulate the problem in a more economic way, with fewer integer variables.

1.3 Mixed integer programming

As we discussed earlier mixed integer programs (MIPs) contain variables that are both integer and continuous. You can apply the branch and bound algorithm the same way as described above for MIPs with the following simple changes:

1. Branching step: You only branch on variables that are integer.
2. Bounding step: You do not round down (up for minimisation) the bound of the LP relaxation since the objective value of the MIP is most likely fractional.
3. Fathoming step: A solution is considered incumbent if it is integer in the integer variables.

Chapter 2

Integer programming: Formulations using binary variables

We illustrate typical usage of binary variables in modelling by giving a number of examples.

2.1 Fixed costs - the big-M method

Let z represent the quantity of product to be manufactured. The cost per unit of production is c. However, this is just the variable production cost. If any of the product is manufactured a fixed cost is incurred which is f. That is:

$$\text{production cost} = \begin{cases} 0 & \text{if } z = 0, \\ f + cz & \text{if } z > 0. \end{cases}$$

Whereas this cost function might seem linear, it is *not*. (Setting the cost to be $f+cz$ everywhere would be linear, however, it would give f for $z = 0$ instead of 0.) We will need a binary variable to express such a cost function.

We wish to minimise the total cost. To model this situation we introduce a new binary $(0-1)$ variable δ, which we will use to represent the logical states: *(a)* no product is produced; *(b)* some amount $z > 0$ is produced. In other words we need to model the following situation:

$$\text{If } z > 0 \text{ then } \delta = 1. \tag{2.1}$$

Equivalently, this can be written as:

$$\text{If } \delta = 0 \text{ then } z = 0. \tag{2.2}$$

To model this logical condition it is necessary to introduce the maximum amount of the product that can be produced, that is the maximum value of z. Let this be M (we do not need the exact maximum value, only an upper bound). We now model this situation by introducing the following constraint:

$$z \leq M\delta. \tag{2.3}$$

Clearly, if $z > 0$ (i.e. some production is undertaken), then δ is forced to 1 ($z > 0$ and $\delta = 0$ would give the contradiction $0 < z \leq 0$). Thus we can now write the total cost (to be minimised) as

$$\text{production cost} = f\delta + cz.$$

Note that because we are minimising cost, the objective will be trying to set $\delta = 0$ if this is possible, so that we don't have to worry about fixed cost being incurred when no production is undertaken, i.e. when $z = 0$ then the minimisation will set $\delta = 0$.

This is a widely applicable modelling strategy, often referred to as the big-M method. We can introduce a binary variable δ to model the relationship (2.1), provided an upper bound M on the possible values of z. Note that the inequality (2.3) does not directly imply that $\delta = 0$ whenever $z = 0$. This is achieved only because we want to minimise $f\delta$ in the objective.

Sometimes we may need to formulate this converse implication directly, as illustrated by the next example.

2.2 A blending problem - the small-m method

We have to blend three ingredients, A, B and C. The extra condition is: if A is included in the blend then B must also be included. Notice that the blend could be made entirely of ingredient B. Let x_A, x_B and x_C denote the proportions of the ingredients. The maximum value of all three variables is 1 (that is 100%). The minimum proportion at which B is considered to be present in the blend is $m = 0.01$.

We may model this situation by introducing a binary variable δ indicating whether A is in the blend or not. The situation is now modelled as follows.

$$\begin{aligned} x_A &\leq \delta \\ x_B &\geq m\delta \\ x_A + x_B + x_C &= 1 \\ x_A, x_B, x_C &\geq 0 \\ \delta &\in \{0, 1\} \end{aligned}$$

If $\delta = 0$ this forces $x_A = 0$, i.e. the blend will be made entirely of products B and C. On the other hand if $\delta = 1$, this forces $x_B \geq m$, i.e. the proportion of B will be at least $m = 0.01$.

Such a formulation is called the small-m method. For a variable z ($z = x_B$ in the example), we aim to express the converse of (2.1), that is

$$\text{If } \delta = 1 \text{ then } z > 0. \tag{2.4}$$

or equivalently,

$$\text{If } z = 0 \text{ then } \delta = 0. \tag{2.5}$$

Analogously to the upper bound M in the previous method, we need a lower bound $m > 0$ which expresses the intended minimal positive value of z. Assuming that we always have $z \geq m$ we can use the inequality:

$$z \geq m\delta.$$

which guarantees:

$$\text{If } \delta = 1 \text{ then } z \geq m \text{ and thus } z > 0. \tag{2.6}$$

Thus, the variable z can only take value 0 or greater than or equal to m. Values of z in $0 \leq z < m$ are neglected.

2.3 Facility location

Let us illustrate the "big-M-method" on a more complex modelling example. A logistic company has to supply a set T of stores from a set F of possible facilities. They can choose an arbitrary set of these facilities to open. For each facility f, there is a significant opening cost b_f they have to pay irrespective to how intensively this facility will be used.

The other part of the cost is transportation. Each store t has to be supplied with r_t units of goods, and the cost of transportation is different between different pairs of facilities and stores: the cost of transporting one unit of goods from facility f to store t is c_{ft}. We must of course supply stores from open facilities.

Let us assign a binary variable δ_f to each facility: $\delta_f = 1$ means this facility is open and $\delta_f = 0$ means it is not. Let the variable x_{ft} express the amount of goods supplied by facility f to store t. A natural upper bound on x_{ft} is r_t as that is the total amount of goods needed in store t. We have the following big-M-type constraint ensuring that we only use open facilities:

$$x_{ft} \leq r_t \delta_f \quad \text{for all } f \in F, t \in T.$$

The entire model can be formulated as follows:

$$\begin{aligned}
\min \sum_{f \in F} b_f \delta_f + \sum_{t \in T} \sum_{f \in F} c_{ft} x_{ft} & & \\
\sum_{f \in F} x_{ft} = r_t & & \text{for all } t \in T, \\
x_{ft} \leq r_t \delta_f & & \text{for all } f \in F, t \in T, \\
x_{ft} \geq 0 & & \text{for all } f \in F, t \in T, \\
\delta_f \in \{0, 1\} & & \text{for all } f \in F.
\end{aligned}$$

Let us now include an additional restriction: every store must choose a facility and has to get its entire supply from that: it is not possible to transport goods from multiple facilities.

This constraint can be included by adding several additional binary variables. Let β_{ft} express that store t is supplied from facility f. First, we need that stores can be only supplied from open facilities. This means that $\beta_{ft} = 1$ should not be allowed if $\delta_f = 0$. This can be added by the simple inequality

$$\beta_{ft} \leq \delta_f \quad \text{for all } f \in F, t \in T.$$

Instead of the previous big-M-type constraints, we must formulate that

$$x_{ft} = \begin{cases} r_t & \text{if } \beta_{ft} = 1 \\ 0 & \text{if } \beta_{ft} = 0. \end{cases}$$

This can be captured by the following constraint.

$$x_{ft} = r_t \beta_{ft} \quad \text{for all } f \in F, t \in T.$$

Finally, we need to make sure that for every store $t \in T$, we will choose to supply it from exactly 1 facility. That is, among the values β_{ft} over all $f \in F$, we must have exactly one 1 and all others 0. This is enforced by

$$\sum_{f \in F} \beta_{ft} = 1 \quad \text{for all } t \in T. \tag{2.7}$$

Let us summarise the model

$$\begin{aligned}
\min \sum_{f \in F} b_f \delta_f + \sum_{t \in T} \sum_{f \in F} c_{ft} x_{ft} & \\
\beta_{ft} \leq \delta_f & \qquad \text{for all } f \in F, t \in T, \\
\sum_{f \in F} \beta_{ft} = 1 & \qquad \text{for all } t \in T \\
x_{ft} = r_t \beta_{ft} & \qquad \text{for all } f \in F, t \in T, \\
x_{ft} \geq 0 & \qquad \text{for all } f \in F, t \in T, \\
\delta_f \in \{0, 1\} & \qquad \text{for all } f \in F, \\
\beta_{ft} \in \{0, 1\} & \qquad \text{for all } f \in F, t \in T.
\end{aligned}$$

Note that the variables x_{ft} are not really necessary and could be replaced simply by $r_t \beta_{ft}$: all old inequalities on them are implied by the new inequalities on the β_{ft} variables. This would leave us with the variables δ_f and β_{ft} only. Indeed, these do describe the entire solution.

2.4 Expressing logical conditions

We used binary variables in the previous examples to express logical relationships. Let us now systematically explore some cases. Consider two possible events X_1 and X_2. and let us associate binary variables x_1 and x_2 to them. $x_i = 1$ means that event X_i takes place, and $x_i = 0$ means that it does not. We can express the basic logical operations the following way:

$$\begin{array}{rr}
x_1 + x_2 \geq 1 & \text{`}X_1 \text{ or } X_2\text{'} \\
x_1 + x_2 = 1 & \text{`either } X_1 \text{ or } X_2\text{'} \\
x_1 = 1, x_2 = 1 & \text{`}X_1 \text{ and } X_2\text{'} \\
x_1 = x_2 & \text{`}X_1 \text{ if and only if } X_2\text{'} \\
x_1 \leq x_2 & \text{`if } X_1 \text{ then } X_2\text{'}
\end{array}$$

Note that 'X_1 or X_2' means that at least one of them or possibly both of the events occur. Whereas 'either X_1 or X_2' means that exactly one of them occurs but not both.

Remarks:

1. Logical conditions often can be transformed into other equivalent logical conditions: 'not (X_1 and X_2)', for example, is equivalent to '(not X_1) or (not X_2)'. Similarly, 'if X_1 then X_2' is equivalent to '(not X_1) or X_2'. We can represent '(not X_1)' by $(1 - x_1)$, thus 'if X_1 then X_2' is represented by $x_1 \leq x_2$, which is equivalent to $(1 - x_1) + x_2 \geq 1$ which is the variable representation of the event '(not X_1) or X_2'.

2. The constraints for the condition 'X_1 and X_2' have been mentioned here because they can be used when an 'and'-condition is part of a larger, more complex logical expression. In the case of a simple 'and'-statement we do not need indicator variables. For constraints stated in LP and IP problems it is assumed that there is an 'and'-relationship between them: we assume the first constraint holds 'and' the second constraint holds 'and' so on. This implies that for an expression like 'X_1 and X_2' it is normally sufficient just to add the expressions X_1 and X_2 as regular constraints.

3. We can generalise this to the case of more than two events and express longer more complicated conditions using binary variables.

Let us now see examples for n events: $X_1, X_2, \ldots, X_n$ with indicator variables $x_1, x_2, \ldots, x_n$. For some $0 \leq k \leq n$ we can use the following inequalities.

$$\begin{array}{lc}
\displaystyle\sum_{i=1}^{n} x_i \leq k & \text{`at most } k \text{ of the events can happen'} \\
\displaystyle\sum_{i=1}^{n} x_i = k & \text{`exactly } k \text{ of the events must happen'} \\
\displaystyle\sum_{i=1}^{n} x_i \geq k & \text{`at least } k \text{ of the events can happen'}
\end{array}$$

If you look at constraint (2.7) of the facility location problem, this is an example where the event that every store has to be supplied from exactly one facility was expressed in a constraint of binary variables.

Example 2.1. As another example, consider the following problem. A clothing company has to decide on the set of products to manufacture. There are four possible types of shirts, A, B, C and D and two types of ties, P and Q. They want to consider manufacturing ties only if they decide to manufacture at least three types of shirts.

If at least one of P and Q is produced, then at least 3 of A, B, C and D must be produced

Let us express this with the corresponding binary variables $x_A, x_B, x_C, x_D, x_P, x_Q$. Let us add one more binary variable y to express that at least one of the ties (P and Q) is being produced. Equivalently, if $x_P + x_Q > 0$, then $y = 1$. This can be done with the big-M method - the maximum value of $x_P + x_Q$ is 2. Hence we can formulate the constraint

$$x_P + x_Q \leq 2y.$$

We then need that if $y = 1$ then $x_A + x_B + x_C + x_D$ is at least 3. We can formulate a constraint analogous to the small-m method, with $m = 3$:

$$3y \leq x_A + x_B + x_C + x_D.$$

The above says that if $y = 1$ then $x_A + x_B + x_C + x_D \geq 3$. These two constraints together express the required relationship.

Remark. It might be tempting to simplify it further by substituting $\frac{1}{2}(x_P + x_Q) \leq y$ from the first inequality, however, it is *not possible.* We would get

$$\frac{3}{2}(x_P + x_Q) \leq x_A + x_B + x_C + x_D. \tag{2.8}$$

Indeed, if exactly one of x_P and x_Q is 1 and the other is 0, the left hand side would be only $\frac{3}{2}$, allowing only 2 of x_A and x_B being 1, that is, producing only 2 shirts. Thus, we need:

$$\frac{3}{2}(x_P + x_Q) \leq 3y \leq x_A + x_B + x_C + x_D. \tag{2.9}$$

Enforcing that y is binary guarantees that when $x_P + x_Q = 1$ we get $y = 1$, which guarantees that the right have side of (2.9) is greater than 3. Without y binary we have (2.8) which fails because it implies that $y = \frac{1}{2}$.

2.5 Non-convex feasible region - Indicator variables for constraints

In the previous example we have seen how binary variables can powerfully model cases when the decision variables may take discrete variables only. Yet we might need to introduce binary variables even in cases when all decision variables are continuous, but the constraints differ from the usual LP constraints. We consider the following example:

Example 2.2. Assume a small beer brewery has to decide how much lager beer and ale they wish to produce. Whereas they are willing to brew both types, the management has the vision of building a strong brand with a dominant product it can be associated with. Either they want to produce more lager than ale by at least 6,000 barrels, or they want to produce more ale than lager by at least 4,000 barrels. They do not wish to produce anything in between, they want to avoid the two amounts to be roughly the same. Also, the maximum amount they can produce in total is 10,000 barrels. If the amount of lager and ale is x_1 and x_2, then constraint they wish to add is:

$$\text{either } x_1 - x_2 \geq 6000 \tag{2.10a}$$

$$\text{or } x_1 - x_2 \leq -4000. \tag{2.10b}$$

The feasible region cannot be modelled by linear constraints. We need to add a binary variable δ which is 1 if production is "lager-dominant" (2.10a) and 0 if it is "ale-dominant" (2.10b). We aim to keep (2.10a) and make (2.10b) void if $\delta = 1$, and conversely, keep (2.10b) and make (2.10a) void if $\delta = 0$. This will be done similarly to the big-M method.

As the total production is at most 10,000, we get the bounds

$$-10000 \leq x_1 - x_2 \leq 10000.$$

Let us consider the following two inequalities.

$$x_1 - x_2 \geq 6000\delta - 10000(1-\delta) \tag{2.11a}$$

$$x_1 - x_2 \leq -4000(1-\delta) + 10000\delta. \tag{2.11b}$$

In case of $\delta = 1$, the first is identical to (2.10a), whereas the second gives

$$x_1 - x_2 \leq 10000.$$

This is void as it is always satisfied due to the constraint on total production. Conversely, if $\delta = 0$, then the first inequality would give the void

$$x_1 - x_2 \geq -10000,$$

whereas the second gives (2.10b). The inequalities (2.11a and b) can be written more concisely as

$$x_1 - x_2 - 16000\delta \geq -10000$$

$$x_1 - x_2 - 14000\delta \leq -4000.$$

In the above example, we formulated dependence of constraints when exactly one of them can occur (either/or relationship), whereas normally we assume that all constraints should occur at the same time (and relationship). We had a linear constraint $ax \leq b$, and a binary variable δ, and we wanted to guarantee that if $\delta = 1$, then $ax \leq b$ is applicable, but otherwise it is not enforced.

Let M denote an upper bound on $ax - b$, i.e. M is the maximum possible violation of this constraint, so the following holds for every solution:

$$ax \leq M \tag{2.12}$$

Then we can formulate the constraint

$$ax \leq b\delta + M(1-\delta).$$

Indeed, if $\delta = 0$ then this constraint turns into (2.12) that is always true. If $\delta = 1$ then it gives back $ax \leq b$, as required.

We might also need to model the converse relationship: whenever $ax \leq b$ is satisfied, then $\delta = 1$ must hold, or equivalently, $\delta = 0$ implies $ax > b$. $ax > b$ is equivalent to $ax \geq b + m$ for some small $m > 0$.

First we need a lower bound, let $M' \geq 0$ denote a value such that

$$ax \geq -M' \tag{2.13}$$

must always hold. We enforce that if $ax \leq m$ then $\delta = 1$ must hold or equivalenty if $\delta = 0$ then $ax \geq b + m$ for some small $m > 0$ by the constraint:

$$ax \geq (b+m)(1-\delta) - M'\delta,$$

or more concisely,

$$ax + (b + m + M')\delta \geq b + m.$$

Indeed, if $\delta = 0$, we obtain the required $ax \geq b + m$ for some small $m > 0$. Further, if $\delta = 1$ then this constraint holds trivially due to (2.13) and nothing is enforced.

Thus, if we want to model the relationship: $\delta = 1$ if and only if $ax \leq b$ we need to write:

$$(b+m)(1-\delta) - M'\delta \leq ax \leq b\delta + M(1-\delta).$$

2.6 Variables with restricted value ranges

Integer variables are appropriate to use when a variable can only take integer values, e.g. the number of machines of a certain type to be purchased. If our variable x should be an integer in $\{\ell, \ell+1, \ldots, u-1, u\}$, we can simply declare x as an integer variable with the inequalities $\ell \leq x$, $x \leq u$.

Consider now the additional requirement that x should be divisible by 3. A useful trick is that instead of x, we introduce an integer variable for $\frac{x}{3}$. That is, we declare y to be integer, and add the inequalities $\ell \leq 3y$, $3y \leq u$. In the formulation, we replace each occurrence of x by $3y$.

In certain cases, we might have some irregular restriction on variables. For example, x should be exactly one of 3, 5, 12, 17, 19: no other values are admissible. This can be formulated by adding binary variables corresponding to the possible values of x. Let $\delta_3, \delta_5, \delta_{12}, \delta_{17}, \delta_{19}$ denote the indicator variables, e.g. $\delta_5 = 1$ if $x = 5$ and 0 otherwise. We need to enforce that exactly one of these is true:

$$\delta_3 + \delta_5 + \delta_{12} + \delta_{17} + \delta_{19} = 1.$$

Then we can define x the following way:

$$x = 3\delta_3 + 5\delta_5 + 12\delta_{12} + 17\delta_{17} + 19\delta_{19}.$$

This will guarantee that x takes exactly one of the prescribed values. These prescribed values do not necessarily need to be integers but they can also be real numbers.