Chapter 11
Introduction to Optimization

Milind G. Sohoni

1 Introduction

Broadly, one may describe **management science** as an interdisciplinary study of problem solving and decision making in human organizations. Management science uses a combination of analytical models and behavioral sciences to address complex business and societal problems. Often, finding a solution involves recognizing the optimization model at the core of the business application, formulating it appropriately, and solving it to gain managerial insights. The classic problems of management science include finding the right mix of inputs to minimize the cost of producing gasoline (blending problem), production planning in a manufacturing setup, inventory and workforce optimization to minimize the expected cost of meeting operating plans for matching supply to demand (aggregate planning), deciding optimal routes to offer for an airline (network planning), assigning crew to manage a schedule (crew scheduling), and maximizing the expected return subject to acceptable levels of variance (portfolio planning). Several textbooks are available that describe (Bazaraa et al., 2011; Bertsimas and Tsitsiklis, 1997; Chvátal, 1983; Wagner, 1969) these and several of other applications. Moreover, almost every chapter in this book includes examples of optimization, such as service level optimization in healthcare analytics and supply chain analytics, portfolio selection in Financial Analytics, inventory optimization in supply chain analytics as well as retail analytics, and price and revenue optimization in pricing analytics. The chapter on simulation provides a glimpse at combining simulation and optimization, for example, when the objective function is difficult to evaluate but can be specified using a simulation model. The case studies in this book illustrate the use of

M. G. Sohoni (✉)
Indian School of Business, Hyderabad, Telangana, India
e-mail: milind_sohoni@isb.edu

B. Pochiraju, S. Seshadri (eds.), *Essentials of Business Analytics*, International Series in Operations Research & Management Science 264,
https://doi.org/10.1007/978-3-319-68837-4_11

optimization in diverse settings: insurance fraud detection, media planning, and airline route planning. Optimization is also embedded in many predictive analytics methods such as forecasting, least square, lasso and logistics regression, maximum likelihood estimation, and backpropagation in neural networks.

It is natural to ask: What is meant by optimality? What is an algorithm? Is there a step-by-step approach that can be used to set up a model? What is the difference between constrained and unconstrained optimization? What makes these problems easy or hard to solve? How to use software to model and solve these problems? There are several books that cover some of these concepts in detail. For example, Bazaraa et al. (2011), Bertsimas and Tsitsiklis (1997), and Luenberger and Ye (1984) contain precise definitions and detailed descriptions of these concepts. In this chapter, we shall illustrate some of the basic ideas using one broad class of optimization problems called linear optimization. Linear optimization covers the most widely used models in business. In addition, because linear models are easy to visualize in two dimensions, it offers a visual introduction to the basic concepts in optimization. Additionally, we also provide a brief introduction to other optimization models and techniques such as integer/discrete optimization, non-linear optimization, search methods, and the use of optimization software.

But before we continue further we briefly touch upon the need to build mathematical models. Representing real-world systems as abstract models, particularly mathematical models, has several advantages. By analyzing the model in a virtual setting, the modeler is able to decide on actions to follow or policies to implement. The modeler is able to gain insights into the complex relationships among many decision variables and inform his/her judgment before selecting/formulating a policy to bring into action.

Figure 11.1 schematically represents the basic idea of model building. A good model abstracts away from reality without losing the essence of the trade-off

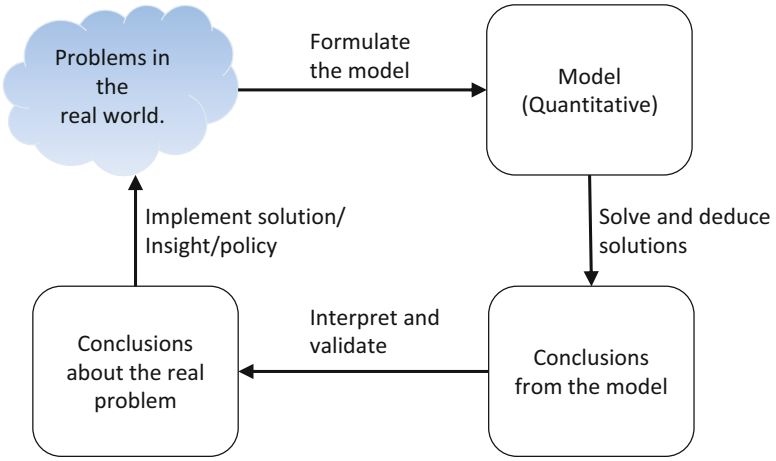


Fig. 11.1 Abstracting from a real-world problem

that needs to be considered. Once the model is analyzed the output needs to be interpreted appropriately and implemented in the real world with suitable modifications.

2 Methods in Optimization: Linear Programming

In this section we focus on a class of analytical models, and their solution techniques, that capture a real-world problem as a mathematical model (linear program) and understand how to interpret their analytical solutions and develop insights. Such mathematical programs are often referred to as (linear) optimization problems (models). The basic idea behind any optimization model is to find the "best" (optimal) solution, with respect to some objective, among all possible (feasible) solutions. While the **objective** depends on the real-world problem that is being modeled, it is represented as a mathematical function capturing the trade-off between the **decisions** that need to be made. The feasible solutions depend on the **constraints** specified in the real-world problem and are also represented by mathematical functions. A general mathematical program then tries to identify an extreme point (i.e., minimum or maximum) of a mathematical function that satisfies a set of constraints. Linear programming (LP) is the specialization of mathematical programming where both, the function—called the **objective function**—and the problem constraints are linear. We explain the notion of linearity in Sect. 2.3. The type of decision variables and constraints depends on the techno-socio-economic nature of the real-world application. However, irrespective of the domain of the application, an important factor for the applicability of optimization methodology is **computational tractability**. With modern computing technology, tractability requirements imply the existence of effective, and efficient, algorithmic procedures that are able to provide a fast solution to these models in a systematic manner.

The Simplex algorithm is one such powerful computational procedure for LPs that is readily applicable to very large-scale applications, sometimes including hundreds of thousands of decision variables. George Dantzig is credited with the development of the Simplex algorithm in 1947, as one of the first mathematical programming algorithms. Today, there are numerous successful implementations[1] of the Simplex algorithm that routinely solve complex techno-socio-economic problems. It is noteworthy that the success of the Simplex method, and the wide application of LP, has led to the broader field of **operations research/management science** being accepted as a scientific approach to decision making.

Over the past few decades, techniques to solve linear optimization problems have evolved significantly. While implementation (and computational speed) of the Simplex algorithm has improved dramatically in most commercially avail-

[1]The reader is referred to http://pubsonline.informs.org/journal/inte (accessed on Jul 22, 2018) to read about several industrial applications of optimization problems.

able solvers, newer mathematical algorithms and implementations have also been developed that compete with the Simplex algorithm effectively. From a business analytics standpoint, however, understanding the models being built to address the optimization problem, the underlying assumptions, and pertinent interpretation of the obtained analytical solutions are equally important. In this chapter, we discuss these details of the linear modeling. We will try to build our understanding using a prototypical LP example in Sect. 2.1 and two-dimensional geometry in Sect. 2.4. The insights gained are valid for higher-dimensional problems too and also reveal how the Simplex algorithm works. For a detailed description of the Simplex algorithm and other solution algorithms the reader is referred to Bazaraa et al. (2011), Bertsimas and Tsitsiklis (1997), and Chvátal (1983).

2.1 A Prototype LP Problem: Glass Manufacturer's Profit

Consider the following problem[2] for a manufacturer who produces two types of glasses, P_1 and P_2. Suppose that it takes the manufacturer 6 h to produce 100 cases of P_1 and 5 h to produce 100 cases of P_2. The production facility is operational for 60 h per week. The manufacturer stores the week's production in her own stockroom where she has an effective capacity of 15,000 ft^3. Hundred cases of P_1 occupy 1000 ft^3 of storage space, while 100 cases of P_2 require 2000 ft^3 due to special packaging. The contribution margin of P_1 is \$5 per case; however, the only customer available will not accept more than 800 cases per week. The contribution of P_2 is \$4.5 per case and there is no limit on the amount that can be sold in the market. The question we seek to answer is the following: *How many cases of each product should the glass manufacturer produce per week in order to maximize the total weekly contribution/profit?*

2.2 The LP Formulation and Solution

It is important to first notice that this is an optimization problem. The objective is to maximize the weekly profit. Furthermore, we are going to maximize the glass manufacturer's weekly profit by adjusting the weekly production levels for P_1 and P_2. Therefore, these weekly production levels are our **control/decision variables**. For ease of representation, we denote our decision variables in hundreds of cases of P_1 and P_2. Let these decision variables be represented as follows:

[2]This example is based on the example described in Chapter 1 of "Applied Mathematical Programming", by Bradley et al. (1977).

x_1:	Number of units (in hundreds of cases) of product P_1 produced weekly, and
x_2:	Number of units (in hundreds of cases) of product P_2 produced weekly

Using these decision variables, we can now represent the manufacturer's objective function analytically as:

$$\max \qquad f\,(x_1,\, x_2) \equiv 500x_1 + 450x_2. \tag{11.1}$$

Equation (11.1) is called the **objective function**, and the coefficients 500 and 450 are called the **objective function coefficients**.

In our problem description, however, the manufacturer is resource constrained, i.e., the manufacturer has limited weekly production and storage capacity. Additionally, the demand for P_1 in the market is limited. Hence, we need to represent these **technological constraints** in our analytical formulation of the problem. First, let's focus on the **production constraint**, which states that the manufacturer has 60 h of production capacity available for weekly production. As mentioned in the problem statement, 100 cases of P_1 require 6 h of production time and that of P_2 require 5 h of production time. The technological constraint imposing this production limitation that our total weekly production doesn't exceed the available weekly production capacity is analytically expressed by:

$$6x_1 + 5x_2 \leq 60. \tag{11.2}$$

Notice that in (11.2) time is measured in hours. Following a similar line of reasoning, the **storage capacity constraint** is analytically represented as:

$$10x_1 + 20x_2 \leq 150. \tag{11.3}$$

From our problem statement, we know that the weekly demand for P_1 does not exceed 800 cases. So we need not produce more than 800 cases of P_1 in the week. Thus, we add a maximum demand constraint as follows:

$$x_1 \leq 8. \tag{11.4}$$

Constraints (11.2), (11.3), and (11.4) are known as the **technological constraints** of the problem. In particular, the coefficients of the variables $x_i, i = 1, 2$, are known as the **technological coefficients** while the values on the **right-hand side** of the three inequalities are referred to as the right-hand side (**rhs**) vector of the constraints.

Finally, we recognize that the permissible value for variables $x_i, i = 1, 2$, must be nonnegative, i.e.,

$$x_i \geq 0\,;\ i = 1, 2, \tag{11.5}$$

since these values express production levels. These constraints are known as the **variable sign restrictions**. Combining (11.1)–(11.5), the LP formulation of our problem is as follows:

$$\begin{aligned} \max \quad & 500x_1 + 450x_2 && (11.6)\\ \text{s.t.} \quad & 6x_1 + 5x_2 \leq 60 && : \text{Production constraint},\\ & 10x_1 + 20x_2 \leq 150 && : \text{Storage capacity constraint},\\ & x_1 \leq 8 && : \text{Max demand for } P_1,\\ & x_1,\ x_2 \geq 0 && : \text{Non-negativity constraint.} \end{aligned}$$

2.3 The General Form of a LP Formulation

In general, a maximization linear programming problem (LPP) can be represented analytically as follows:

Objective function:

$$\max \qquad f(x_1, x_2, \ldots, x_n) \equiv c_1x_1 + c_2x_2 + \cdots + c_nx_n \tag{11.7}$$

s.t. Technological constraints:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \begin{pmatrix} \leq \\ = \\ \geq \end{pmatrix} b_i, \quad i = 1, \ldots, m, \tag{11.8}$$

Sign restrictions:

$$(x_j \geq 0) \text{ or } (x_j \leq 0) \text{ or } (x_j \text{ is unrestricted}), \quad j = 1, \ldots, n. \tag{11.9}$$

The formulation (11.7)–(11.9) has the general structure of a mathematical programming problem. Moreover, there is a specific structure to this formulation, i.e., the functions involved in the problem objective and the left-hand side (lhs) of the technological constraints are **linear**. It is the assumptions implied by linearity that to a large extent determine the applicability of the above model in real-world applications. To understand this concept of linearity a bit better, assume that the different decision variables $x_1, \ldots, x_n$ correspond to various activities from which any solution is eventually constructed. Essentially, the assigned values in a solution indicate the activity level in the plan considered. Each technological constraint of (11.8) imposes some restriction on the consumption of a particular resource (similar to the production and storage resources described in the prototype example (11.6).) Under this interpretation, the linearity property implies the following:

Additivity assumption: The total consumption of each resource and the overall objective value are the aggregates of the resource consumptions and the contributions to the problem objective, resulting by carrying out each activity independently.

Proportionality assumption: The consumptions and contributions for each activity are proportional to the actual activity level.

Divisibility assumption: Each variable is allowed to have fractional values (continuous variables).

Certainty assumption: Each coefficient of the objective vector and constraint matrix is known with certainty (not a random variable).

It is informative to understand how we implicitly applied this logic when we derived the technological constraints of the prototype example: (1) Our assumption that the processing of each case of P_1 and P_2 required constant amounts of time, respectively, implies proportionality, and (2) the assumption that the total production time consumed in the week is the aggregate of the manufacturing times required for the production of each type of glass, if the corresponding activity took place independently, implies additivity.

It is important to note how the linearity assumption restricts our modeling capabilities in the LP framework: For example, we cannot immediately model effects like economies of scale in the problem structure, and/or situations in which resource consumption of resources by complementary activities takes place. In some cases, one can approach these more complicated problems by applying some **linearization** scheme—but that requires additional modeling effort.

Another approximation, implicit in many LPPs, is the so-called **divisibility assumption**. This assumption refers to the fact that for LP theory and algorithms to work, the decision variables must be **real** valued. However, in many business problems, we may want to restrict values of the decision variables to be integers. For example, this may be the case with the production of glass types, P_1 and P_2, in our prototype example or production of aircraft. On the other hand, continuous quantities, such as tons of steel to produce and gallons of gasoline to consume, are divisible. That is, if we solved a LPP whose optimal solution included the consumption of 3.27 gallons of gasoline, the answer would make sense to us; we are able to consume fractions of gallons of gasoline. On the contrary, if the optimal solution called for the production of 3.27 aircraft, however, the solution probably would not make sense to anyone.

Imposing integrality constraints for some, or all, variables in a LPP turns the problem into a **(mixed) integer programming** (MIP or IP) problem. The computational complexity of solving an MIP problem is much higher than that of a LP. Actually, MIP problems belong to the notorious class of *NP-complete* problems, i.e., those problems for which there is no known/guaranteed polynomial bound on the solution time to find an optimal solution. We will briefly discuss the challenge of solving MIPs later in Sect. 3.

Finally, before we conclude this discussion, we define the **feasible region of the LP** of (11.7)–(11.9), as the entire set of vectors $\langle x_1, x_2, \ldots, x_n \rangle$ (notice that each

variable is a coordinate of an n-dimensional vector) that satisfy the technological constraint (11.8) and the sign restrictions (11.9). An **optimal solution** to the problem is any feasible vector that further satisfies the optimality requirement defined by (11.7).

Before we conclude this section, it is worth mentioning that there are several commercially available software packages such as Microsoft Excel's Solver or XLMiner add-ins, Python, SAS, GAMS, etc. to solve LPPs. However, many business applications can be easily modeled using Microsoft Excel's **Solver** add-in program because of the convenient spreadsheet interface available. There are a few online tutorials available to understand how to input a LP model in Solver. We provide a few weblinks in Appendix. We do not describe the steps involved in building a spreadsheet-based LP model in Excel Solver. However, in Appendix section "Spreadsheet Models and Excel Solver Reports" we describe the output generated from Excel Solver, which is closely linked to the discussion provided in the rest of this chapter.

2.4 Understanding LP Geometry

In this section, we develop a solution approach for LP problems, which is based on a geometrical representation of the feasible region and the objective function. While we will work with our prototype example with two decision variables, the insights we gain through this exercise will readily carry over to LPPs with n variables. The number of decision variables in a LP determines the problems **dimensionality**. A two-dimensional (2-D) problem can be represented in a Cartesian coordinate system (two-dimensional space with axes perpendicular to each other) and problems with n variables can be represented by n-dimensional spaces based on a set of n mutually perpendicular axes. In particular, the n-dimensional space (think coordinates) to be considered has each dimension defined by one of the LP variables x_j. While a 2-D problem is easy to represent and visualize, to maintain sanity, it is advisable not to visualize higher-dimensional spaces beyond three dimensions.

As will be shown later with our prototype example, the objective function, in an n-dimensional space, is represented by its **contour plots**, i.e., the sets of points that correspond to the same objective value. As mentioned earlier, to facilitate the visualization of the concepts involved, we shall restrict ourselves to the two-dimensional case. To the extent that the proposed approach requires the visualization of the underlying geometry, it is applicable only for LPs with up to three variables.

2.4.1 Feasible Region for Two-Variable LPs

The primary idea behind the geometrical representation is to correspond every vector $\langle x_1, x_2 \rangle$, denoting the decision variables of a two-variable LP, to the point with coordinates (x_1, x_2) in a two-dimensional (planar) Cartesian coordinate

system. Remember that the set of constraints determine the feasible region of the LP. Thus, under aforementioned correspondence, the feasible region is depicted by the set of points that satisfy the LP constraints and the sign restrictions simultaneously. Since all constraints in a LPP are expressed by linear inequalities, we must first characterize the set of points that constitute the solution space of each linear inequality. The intersection of the solution spaces corresponding to each technological constraint and/or sign restriction will represent the LP feasible region. Notice that a constraint can either be an equality or an inequality in LPP. We first consider the feasible region corresponding to a single equality constraint.

The Feasible Space of a Single Equality Constraint Consider an equality constraint of the type

$$a_1 x_1 + a_2 x_2 = b \tag{11.10}$$

Assuming $a_2 \neq 0$, this equation corresponds to a **straight line** with **slope** $s = \frac{a_1}{a_2}$ and **intercept** $d = \frac{b}{a_2}$. In the special case where $a_2 = 0$, the feasible space *locus* of (11.10) is still a straight line perpendicular to the x_1-axis, intersecting it at the point $\left(\frac{b}{a_1}, 0\right)$. It is noteworthy that an equality constraint restricts the dimensionality of the feasible space by *one degree of freedom*, i.e., in the case of a 2-D problem, it turns the feasible space from a planar area to a line segment.

The Feasible Space of a Single Inequality Constraint Consider the constraint:

$$a_1 x_1 + a_2 x_2 \begin{pmatrix} \leq \\ \geq \end{pmatrix} b \tag{11.11}$$

The feasible space is one of the closed **half-planes** defined by the equation of the line corresponding to this inequality: $a_1 x_1 + a_2 x_2 = b$. Recollect that a line divides a 2-D plane into two halves (half-planes), i.e., the portion of the plane lying on each side of the line. One simple technique to determine the half-plane comprising the feasible space of a linear inequality is to test whether the point $(0, 0)$ satisfies the inequality. In case of a positive answer, the **feasible space** is the half-space containing the origin. Otherwise, it is the half-space lying on the other side which does not contain the origin.

Consider our prototype LP Sect. 2.1 described earlier. Figure 11.2 shows the feasible regions corresponding to the individual technological and nonnegativity constraints. In particular, Fig. 11.2c shows the entire feasible region as the intersection of the half-spaces of the individual constraints. Note that, for our prototype problem, the feasible region is bounded on all sides (the region doesn't extend to infinity in any direction) and nonempty (has at least one feasible solution).

Infeasibility and Unboundedness Sometimes, the constraint set can lead to an **infeasible** or **unbounded** feasible region.

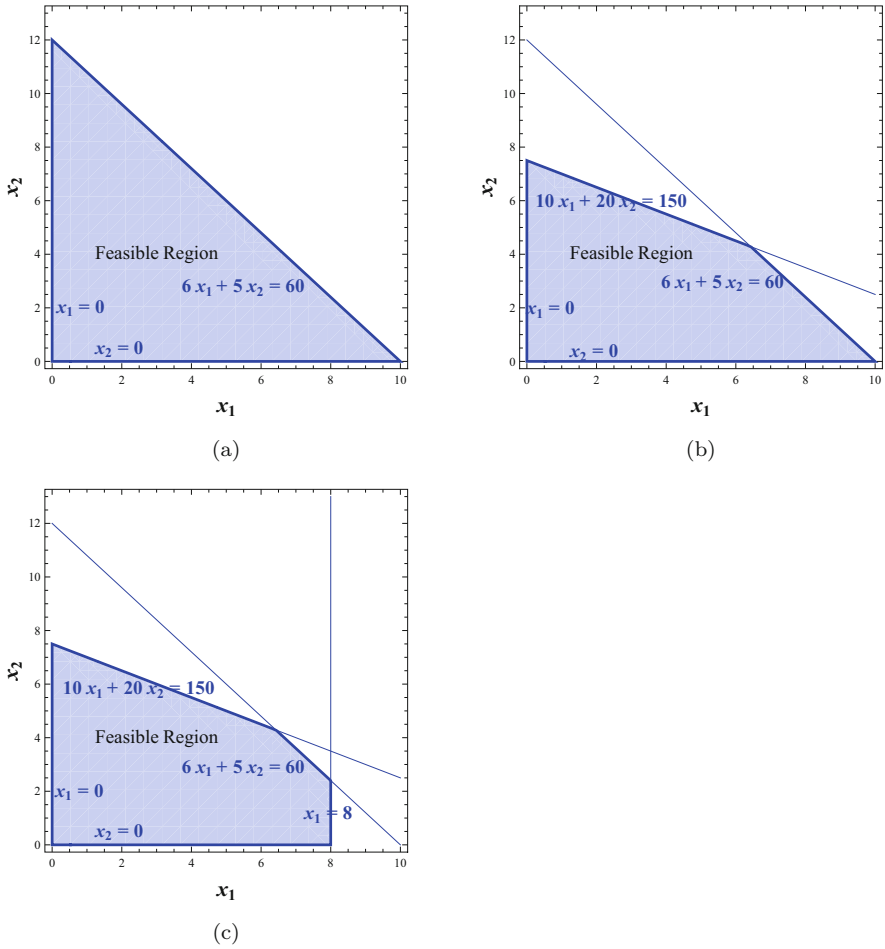


Fig. 11.2 Feasible region of the prototype LP in 2-D. (**a**) Feasible region of the production and nonnegative constraints. (**b**) Feasible region of the storage and production constraint. (**c**) The entire feasible region

An infeasible region implies the constraints are "contradictory" and hence the intersection set of the half-spaces is empty. An unbounded feasible region may mean that the optimal solution could go off to $-\infty$ or $+\infty$ if the objective function "improves" in the direction in which the feasible region is unbounded.

Consider again our original prototype example. Suppose there is no demand restriction on the number of cases of P_1 and the manufacturer requires that at least 1050 cases of P_1 are produced every week. These requirements introduce two new constraints into the problem formulation, i.e.,

$$x_1 \geq 10.5.$$

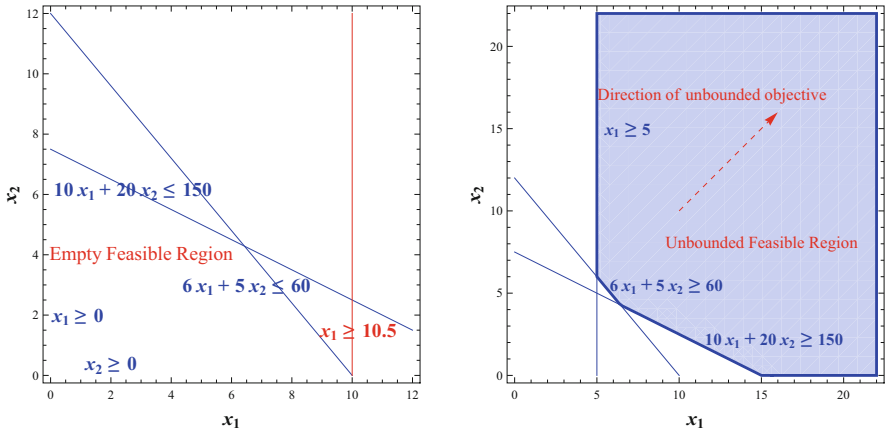


Fig. 11.3 Infeasible and unbounded feasible regions in 2-D. (**a**) Feasible region is empty (Infeasible). (**b**) Feasible region is unbounded

Figure 11.3a shows the feasible region for this new problem which is empty, i.e., there are no points on the (x_1, x_2)-plane that satisfy all constraints, and therefore our problem is infeasible (over-constrained).

To understand unbounded feasible regions visually, consider a situation wherein we change our prototype LP such that the manufacturer must use at least 60 h of production, must produce at least 500 cases of P_1, and must use at least 15,000 units of storage capacity. In this case the constraint set changes to

$$\begin{aligned} x_1 &\geq 5, \\ 6x_1 + 5x_2 &\geq 60, \\ 10x_1 + 20x_2 &\geq 150, \end{aligned}$$

and the feasible looks like the region depicted in Fig. 11.3b. It is easy to see that the feasible region of this problem is unbounded, Furthermore, in this case our objective function, $500x_1 + 450x_2$ can take arbitrarily large values and there will always be a feasible production decision corresponding to that arbitrarily large profit. Such a LP is characterized as **unbounded**. It is noteworthy, however, that even though an unbounded feasible region is a necessary condition for a LP to be unbounded, it is not sufficient (e.g., if we were to minimize our objective function, we would get a finite value).

Representing the Objective Function A function of two variables $f(x_1, x_2)$ is typically represented as a surface in an (orthogonal) three-dimensional space, where two of the dimensions correspond to the independent variables x_1 and x_2, while the third dimension provides the objective function value for any pair (x_1, x_2). In

the context of our discussion, however, we will use the concept of **contour plots**. Suppose α is some constant value of the objective function, then for any given range of α's, a contour plot depicts the objective function by identifying the set of points (x_1, x_2) such that $f(x_1, x_2) = \alpha$. The plot obtained for any fixed value of α is a contour of the function. Studying the structure of a contour identifies some patterns that depict useful properties of the function. In the case of 2-D LPPs, the linearity of the objective function implies that any contour can be represented as a straight line of the form:

$$c_1 x_1 + c_2 x_2 = \alpha. \tag{11.12}$$

It is noteworthy that for a maximization (minimization) problem, this starting line is sometimes referred to as an **isoprofit** (**isocost**) line. Assuming that $c_2 \neq 0$ (o.w., work with c_1), (11.12) can be rewritten as:

$$x_2 = -\frac{c_1}{c_2} x_1 + \frac{\alpha}{c_2}. \tag{11.13}$$

Consider the objective function $500x_1 + 450x_2$ in our prototype example. Let us draw the first isoprofit line as $500x_1 + 450x_2 = \alpha$ (the dashed red line in Fig. 11.4), where $\alpha = 1000$ and superimpose it over our feasible region. Notice that the intersection of this line with the feasible region provides all those production decisions that would result in a profit of exactly \$1000.

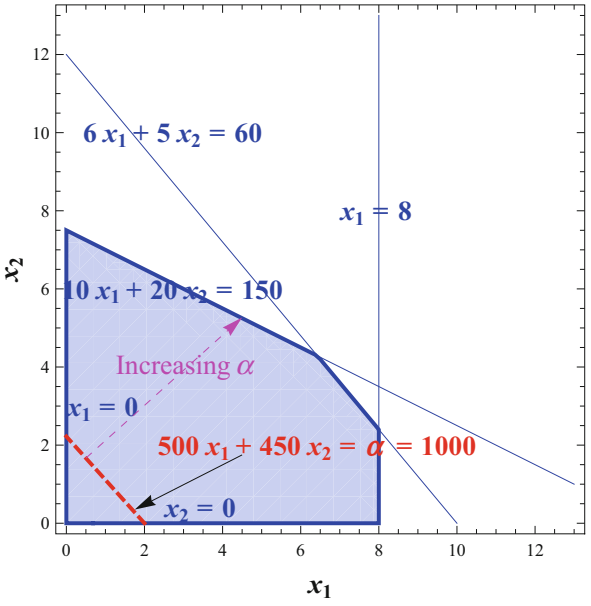


Fig. 11.4 Drawing isoprofit lines over the feasible region

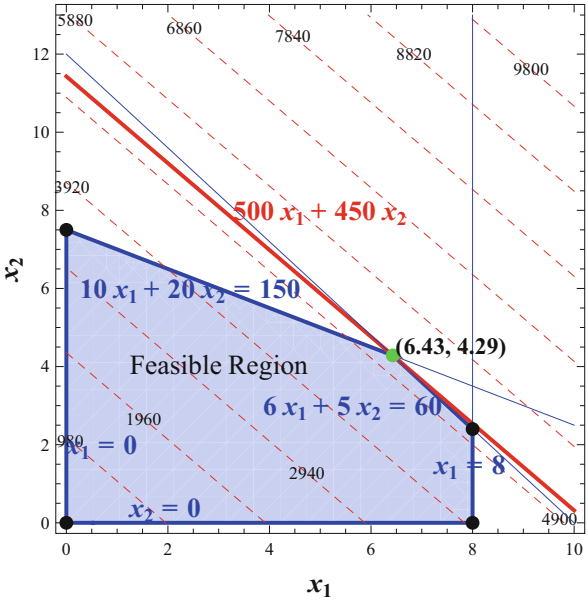


Fig. 11.5 Sweeping the isoprofit line across the feasible region until it is about to exit. An optimal solution exists at a corner point (vertex)

As we change the value of α, the resulting isoprofit lines have constant slope and varying intercept, i.e., they are parallel to each other (since by definition isoprofit/isocost lines cannot intersect). Hence, if we continuously increase α from some initial value α_0, the corresponding isoprofit lines can be obtained by "sliding" the isoprofit line corresponding to $f(x_1, x_2) = \alpha_0$ parallel to itself, in the direction of increasing (decreasing) intercepts, if c_2 is positive (negative.) This "improving direction" of the isoprofit line is denoted by the dashed magenta arrow in Fig. 11.4.

Figure 11.5 shows several isoprofit lines, superimposed over the feasible region, for our prototype problem.

Finding the Optimal Solution It is easy to argue that an optimal solution to a LPP will never lie in the interior of the feasible region. To understand why this must be true, consider the prototype example and let us assume that an optimal solution exists in the interior. It is easy to verify that by simply increasing the value of either x_1 or x_2, or both—as long as we remain feasible—we can improve the objective value. But this would contradict the fact that the point in the interior is an optimal solution. Thus, we can rule out the possibility of finding an optimal solution in the interior of the feasible region. So then, if an optimal solution exists, it must lie somewhere on the boundary of the feasible region. The "sliding motion" described earlier suggests a way for finding the optimal solution to a LPP. The basic idea is to keep sliding the isoprofit line in the direction of increasing α's, until we cross (or are just about to slide beyond) the boundary of the LP feasible region. For our prototype

LPP, this idea is demonstrated in Fig. 11.5. The dashed red lines are the contour lines and the solid red line is the contour line corresponding to that value of α such that any further increase would result in the objective line crossing the feasible region, i.e., an infinitesimal increase in α would result in the contour line moving parallel to itself but not intersecting the feasible region. Thus, the objective value is maximized at that point on the boundary beyond which the objective function crosses out of the feasible region. In this case that point happens to be defined by the intersection of the constraint lines for the production capacity and storage capacity, i.e., $6x_1+5x_2 = 60$ and $10x_1 + 20x_2 = 150$. The coordinates of the optimal point are $x_1^\star = 6.43$ and $x_2^\star = 4.29$. The maximal profit is $f\left(x_1^\star, x_2^\star\right) = 5142.86$.

In fact, notice that the optimal point (the green dot) is one of the **corner** points (the black dots) of the feasible region depicted in Fig. 11.5 and is **unique**. The optimal corner point is also referred to as the **optimal vertex**.

In summary, if the optimal vertex is uniquely determined by a set of intersecting constraints and the optimal solution only exists at that unique corner point (vertex), then we have a unique optimal solution to our problem. See Fig. 11.6.

LPs with Many Optimal Solutions A natural question to ask is the following: Is the optimal solution, if one exists, always unique? To analyze this graphically, suppose the objective function of our prototype problem is changed to

$$225x_1 + 450x_2.$$

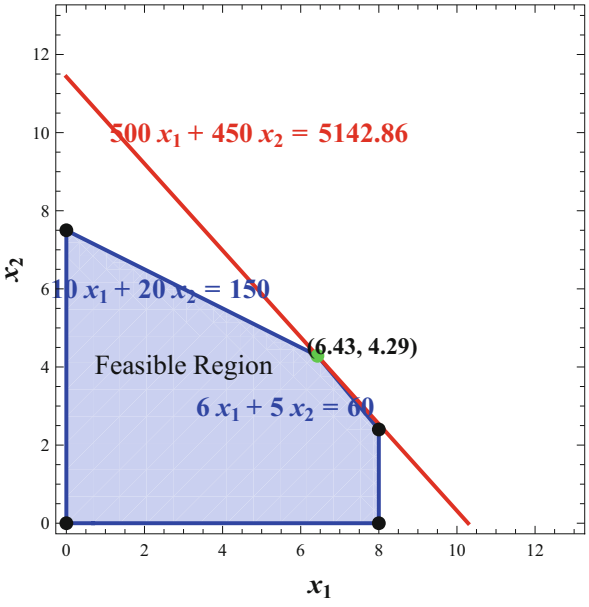


Fig. 11.6 A unique optimal solution exists at a single corner point (vertex)

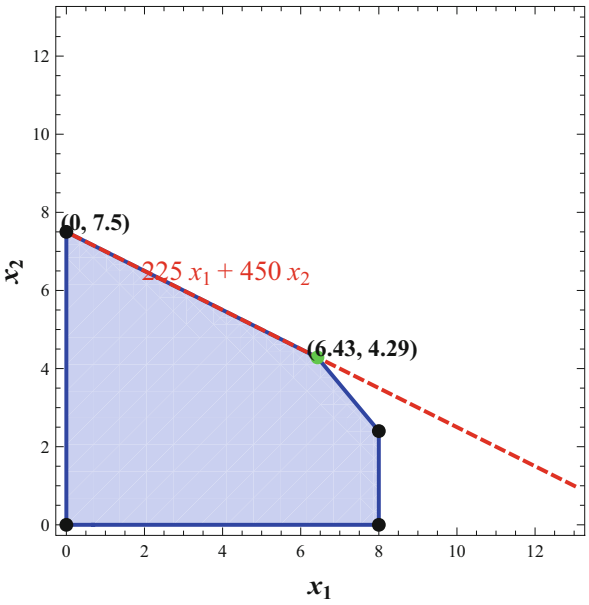


Fig. 11.7 Multiple optimal solutions along a face (includes corner points)

Notice that any isoprofit line corresponding to the new objective function is parallel to the line corresponding to the storage constraint:

$$10x_1 + 20x_2 = 150.$$

Therefore, if we try to apply the graphical optimizing technique described earlier, we get the situation depicted in Fig. 11.7, i.e., every point in the line segment between points (0,7.5) and (6.43, 4.29), along the storage constraint line, is an optimal point, providing the optimal objective value of $5142.86.

It is worth noticing that even in this case of many optimal solutions, we have two of them corresponding to "corner" points of the feasible region, namely, points (0,7.5) and (6.43, 4.29).

Summarizing the above discussion, we have shown that a two-variable LP can either have a unique optimal solution that corresponds to a "corner" point of the feasible region or have many optimal solutions that correspond to an entire "edge" of the feasible region, or be unbounded, or be infeasible. **This is true for general n-dimensional LPPs too**. There is a famous theorem, called the **fundamental theorem of linear programming**, which states that if a LP has a bounded optimal solution, then it must have one that is an extreme point of its feasible region. The **Simplex algorithm** (a solution algorithm embedded in most software) essentially exploits this fundamental result to reduce the space to be searched for an optimal solution.

The fundamental theorem of linear programming states the following:

Theorem 1 (The Fundamental Theorem of Linear Programming) *If a LP has a bounded optimal solution, then there exists an extreme point of the feasible region that is optimal.*

Another important fact about the feasible region of a LP is that it is **convex**. A convex set is defined as follows: Let y_1 and y_2 be any two points belonging to a set S. Then S is a convex set if and only if all points y, belonging to the line segment joining y_1 and y_2, also belong to the set S. In mathematical terms y can be expressed as $y = \alpha y_1 + (1-\alpha) y_2$, for all values of $\alpha \in [0, 1]$, and the set S is a convex set if y also belongs to the set S. An example of a convex set is a circle in two dimensions. The feasible region of a LP is **polyhedral** because it is defined by linear equalities.

2.4.2 Binding Constraints, LP Relaxation, and Degeneracy

A **constraint is binding** if it passes through the optimal vertex, and **nonbinding** if it does not pass through the optimal vertex. If we increase the rhs value of a $\leq$ constraint (see Fig. 11.8a) or decrease the rhs value of a $\geq$ constraint, we "relax" the constraint, i.e., we enlarge the feasible region to include additional points that simultaneously satisfy all the constraints of the original LP. **Relaxing** the LP can only "improve" optimal objective value, i.e., the inclusion of additional feasible points in the feasible region does not remove any of the original feasible points (including the original optimal solution). All that can happen is that one of the new feasible points may provide a better objective function value. On the other hand,

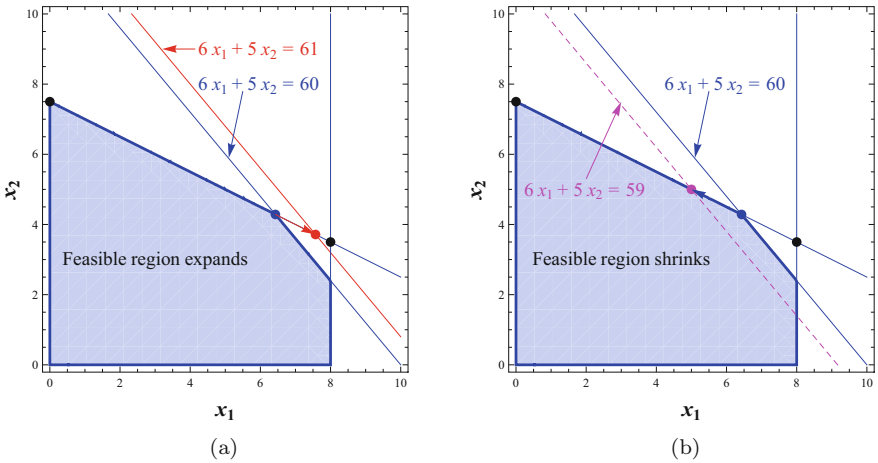


Fig. 11.8 Relaxing and tightening the feasible region. (**a**) Relaxing (expanding) the feasible region. (**b**) Tightening (shrinking) the feasible region

if we **tighten the LP** by decreasing the rhs value of a $\leq$ constraint or increase the rhs value of a $\geq$ constraint (see Fig. 11.8b), the optimal objective value can only deteriorate. In Fig. 11.8 we demonstrate how the feasible region expands (shrinks) as we increase (decrease) the rhs value of the production constraint in our prototype LP.

If a constraint is binding, changing its rhs will change the optimal solution and the objective value. In general, as the rhs of the binding constraint is changed, the optimal vertex *slides along* the intersection of changing constraint and, as the optimal vertex moves, the optimal objective value changes (unless the corner point or vertex is degenerate, which we will discuss later). If a constraint is not binding, then tightening it (a bit) or relaxing it (as much as you please) will not change the optimal solution or the optimal objective value.

A **slack variable** for nonbinding $\leq$ constraint is defined to be the difference between its rhs and the value of the left-hand side of the constraint evaluated at the optimal vertex (in the n-dimensional space). Suppose the optimal vertex is represented by $\langle x_1^*, x_2^*, \ldots, x_n^* \rangle$. Formally, the slack s_i of the ith $\leq$ constraint is defined as

$$s_i = b_i - \sum_{j=1}^{n} a_{ij} x_j^*.$$

Similarly, the **surplus** associated with a nonbinding $\geq$ constraint is the extra value that may be reduced from the constraint's left-hand side function before the constraint becomes binding, i.e., the left-hand side equals the rhs. The formal definition of the surplus variable of an ith $\geq$ constraint is:

$$\text{surplus}_i = \sum_{j=1}^{n} a_{ij} x_j^* - b_i.$$

Any LP involving inequality constraints can be converted into an equivalent LP involving just equality constraints (simply add slack and surplus variables). After such a conversion, the LP formulation can be written as (here we consider only a maximization problem and assume that the constraints are of $\leq$ type):

$$\begin{aligned}
&\max && \textstyle\sum_{i=1}^{n} c_i x_i \\
&\text{s.t.} && \\
&&& \begin{pmatrix}
a_{11}x_1 & +\cdots & +a_{1n}x_n & +s_1 & +0 & +0 & +\cdots & +0 & = b_1 \\
a_{21}x_1 & +\cdots & +a_{2n}x_n & +0 & +s_2 & +0 & +\cdots & +0 & = b_2 \\
\vdots & \vdots & \vdots & \vdots & +0 & +\vdots & +s_i & +\vdots & \vdots \quad \vdots \\
a_{m1}x_1 & +\cdots & +a_{mn}x_n & +0 & +0 & +0 & \cdots & +s_m & = b_m
\end{pmatrix}, \\
&&& x_j \geq 0 \quad j = 1, \ldots, n
\end{aligned}$$

with m equality constraints and n variables (where we can assume $n > m$). Then, theory tells us that each vertex of the feasible region of this LP can be found by: choosing m of the n variables (these m variables are collectively known as the **basis** and the corresponding variables are called **basic variables**); setting the remaining $(n - m)$ variables to zero; and solving a set of simultaneous linear equations to determine values for the m variables we have selected. Not every selection of m variables will give a nonnegative solution. Also, enumerating all possible solutions can be very tedious, though, there are problems where, if m were small, the enumeration can be done very quickly. Therefore, the Simplex algorithm tries to find an "adjacent" vertex that improves the value of the objective function. There is one problem to be solved before doing that: if these values for the m variables are all >0, then the vertex is **nondegenerate**. If one or more of these variables is zero, then the vertex is **degenerate**. This may sometimes mean that the vertex is **over-defined**, i.e., there are more than necessary binding constraints at the vertex. An example of a degenerate vertex in three dimensions is

$$x_1 + 4x_3 \leq 4$$
$$x_2 + 4x_3 \leq 4$$
$$x_1, x_2, x_3 \geq 0$$

The three-dimensional feasible region looks like the region in Fig. 11.9. Notice that the vertex $(0,0,1)$ has four planes defining it.

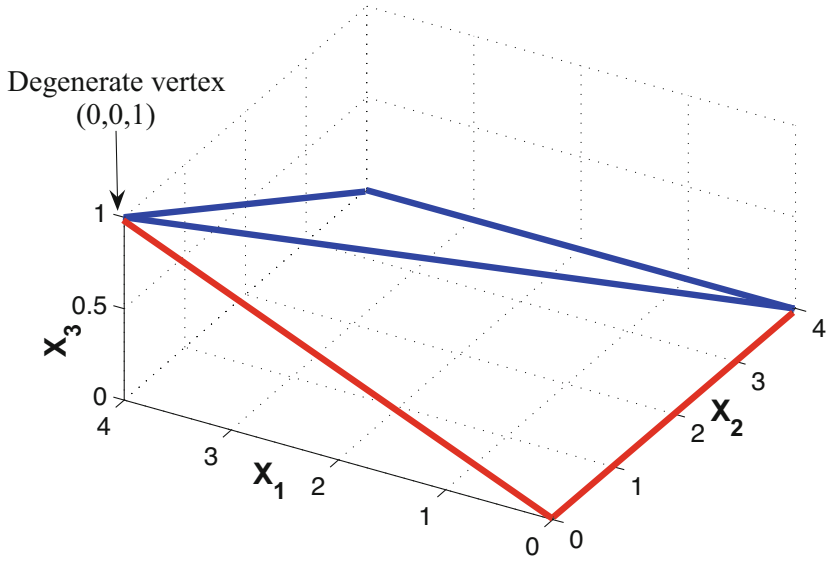


Fig. 11.9 An example of a degenerate vertex

For two-dimensional problems degeneracy is **not** an issue—if there are three constraints binding at a vertex, then one of them is redundant, i.e., one of the constraints can be expressed as a linear combination of the other two and hence removed (not considered). However, in more than two dimensions it may not be possible to eliminate any of the constraints because they may be linearly independent. Consequently, the Simplex algorithm may get stuck at a degenerate vertex (cycling) and may not move to an adjacent vertex that improves the objective value during execution. There are special methods to overcome this problem with degeneracy allowing the Simplex algorithm to break out of cycling at a degenerate vertex.

2.5 Shadow Prices, Reduced Costs, and Sensitivity Analysis

The **shadow price**, associated with a particular constraint, is the **change in the optimal value of the objective function per unit increase in the rhs value for the constraint, all other problem data remaining unchanged**. Equivalently, the shadow price is also the rate of deterioration in the objective value obtained by restricting that constraint. Shadow prices are also called **dual values**. Shadow price is discussed in detail in Chap. 23 on Pricing Analytics.

The **reduced cost** associated with the nonnegativity constraint for each variable is the shadow price of that constraint, i.e., the corresponding change in the objective function per unit increase in the lower bound of the variable. Algebraically, we can express the reduced cost of activity j as

$$\bar{c}_j = c_j - \sum_{i=1}^{m} a_{ij} y_i$$

where c_j is the objective coefficient of activity j, y_i is the shadow price (dual value) associated with constraint i, and a_{ij} is the amount of resource i (corresponds to constraint i) used per unit of activity j. The operation of determining the reduced cost of an activity, j, from the shadow prices of the constraints and the objective function is generally referred to as **pricing out an activity**. To understand these computations, consider the prototype LP described earlier. Suppose the manufacturer decides to add another set of glasses, P_3, to his product mix. Let us assume that P_3 requires 8 h of production time per 100 cases and occupies 1000 cubic units of storage space. Further, let the marginal profit from a case of P_3 be \$6. If x_3 represents the decision of how many hundreds of cases of P_3 to produce, then the new LP can be rewritten as:

max	500	x_1	+450	x_2	+600	x_3			Shadow prices at optimality↓
Production:	6	x_1	+5	x_2	+8	x_3	$\leq$	$b_1 = 60$	: $y_1^\star = 78.57$
Storage:	10	x_1	+20	x_2	+10	x_3	$\leq$	150,	: $y_2^\star = 2.86$
Demand:	1	x_1					$\leq$	8,	: $y_3^\star = 0$
Sign restrictions:		x_1,		x_2,		x_3	$\geq$	0.	

Suppose, we solve this LP to find the following optimal solution:

Optimal obj. value	= 5142.86		
Decision variables ↓		Shadow prices ↓	
$x_1^\star$	= 6.43	First constraint: $y_1^\star$ (***binding***)	= 78.57
$x_2^\star$	= 4.29	Second constraint: $y_2^\star$(***binding***)	= 2.86
$x_3^\star$	= 0	Third constraint: $y_3^\star$ (***non-binding***)	= 0

The **reduced cost of variable** x_1 (here $j = 1$) **at optimality**, i.e., $\bar{c}_1$, is computed as follows, where $c_1 = 500$, $a_{11} = 6$, $a_{21} = 10$, $a_{31} = 1$.

$$\begin{aligned}\bar{c}_1 &= c_1 - \left(a_{11}y_1^\star + a_{21}y_2^\star + a_{31}y_3^\star\right), \\ &= 500 - (6 \times 78.57 + 10 \times 2.86 + 1 \times 0), \\ &= 0.\end{aligned}$$

Similarly, for variable x_3 ($j = 3$) the reduced cost is (where $c_3 = 600$, $a_{13} = 8$, $a_{23} = 10$, $a_{33} = 0$)

$$\begin{aligned}\bar{c}_3 &= c_3 - \left(a_{13}y_1^\star + a_{23}y_2^\star + a_{33}y_3^\star\right), \\ &= 600 - (8 \times 78.57 + 10 \times 2.86 + 0 \times 0), \\ &= -57.14.\end{aligned}$$

Now, suppose we want to compute the **shadow price** of the production constraint. Let b_1 denote the rhs of the production constraint (C1). Currently, $b_1 = 60$ as stated in the formulation above. Notice that the current optimal objective value is 5142.86 when $b_1 = 60$. Let us define the optimal value as a function of rhs of the production constraint, i.e., b_1 and denote it as $Z^\star(b_1)$. Thus, $Z^\star(60) = 5142.86$. Now suppose we keep all other values the same (as mentioned in the formulation) but change b_1 to 61 and recompute the optimal objective value. Upon solving the LP we get the new optimal objective value of $Z^\star(61) = 5221.43$. Then, using the definition of

the shadow price of a constraint, the shadow price of the production constraint is computed as follows:

$$\begin{aligned}\text{Shadow price of C1} &= \frac{Z^{\star}(61) - Z^{\star}(60)}{61 - 60} \\ &= \frac{5221.43 - 5142.86}{1} \\ &= 78.57.\end{aligned}$$

Notice that the shadow price is the **rate at which the optimal objective changes with respect to the rhs of a particular constraint all else remaining equal.** It **should not** be interpreted as the absolute change in the optimal objective value.

Notice two important facts: (1) The *reduced cost of basic variables is 0*, i.e., $\bar{c}_j$ equals 0 for all basic x_j (see Sect. 2.4.2 for the definition), and (2) Since c_j equals zero for slack and surplus variables (see Sect. 2.4.2 for definition) the reduced cost of these variables is always the negative of the shadow price corresponding to the respective constraints. The *economic interpretation of a shadow price*, y_i (associated with resource i), is the *imputed value* of resource i. The term $\sum_{i=1}^{m} a_{ij} y_i$ is interpreted as the total value of the resource used per unit activity j. It is thus the *marginal resource cost* for using that activity. If we think of the objective coefficients c_j as being the marginal revenues, the reduced costs, $\bar{c}_j$, are simply the *net marginal revenues*.

An intuitive way to think about reduced costs is as follows: If the optimal solution to a LP indicates that the optimal level of a particular decision variable is zero, it must be because the objective function coefficient of this variable (e.g., its unit contribution to profits or unit cost) is not beneficial enough to justify its "inclusion" in the decision. The reduced cost of that decision variable tells us the amount by which the objective function coefficients must improve for the decision variable to become "attractive enough to include" and take on a nonzero value in the optimal solution. Hence the reduced costs of all decision variables that take nonzero values in the optimal solution are, by definition, zero $\Rightarrow$ no further enhancement to their attractiveness is needed to get the LP to use them, since they are already "included." In economic terms, the values imputed to the resources (x_j) are such that the *net marginal revenue* is zero on those activities operated at a positive level, i.e., *marginal revenue* = *marginal cost* (MR = MC).

Shadow prices are only locally accurate (shadow prices are valid over a particular range, i.e., as long as the *set of binding constraints does not change* the shadow price of a constraint remains the same.); if we make dramatic changes in the constraint, naively multiplying the shadow price by the magnitude of the change may mislead us. In particular, the shadow price holds only within an allowable range of changes to the constraints rhs; outside of this allowable range the shadow price may change. This allowable range is composed of two components. The **allowable increase** is the amount by which the rhs may be increased before the shadow price can change; similarly, the **allowable decrease** is the corresponding

reduction that may be applied to the rhs before a change in the shadow price can take place (whether this increase or decrease corresponds to a tightening or a relaxation of the constraint depends on the direction of the constraints inequality). **A constraint is binding** if it passes through the optimal vertex, and **nonbinding** if it does not pass through the optimal vertex (constraint C3 in the example above). For a binding constraint, the geometric intuition behind the definition of a shadow price is as follows: By changing the rhs of a binding constraint, we change the optimal solution as it slides along the *other binding constraints*. Within the allowable range of changes to the rhs, the optimal vertex slides in a straight line, and the optimal objective value changes at a constant rate (which is the shadow price). Once we cross the limits indicated by the allowable increase or decrease, however, the optimal vertex's slide changes because **the set of binding constraints change**. At some point the constraint, whose rhs is being modified, may become nonbinding and a new vertex is optimal. For a nonbinding constraint the shadow price (or dual value) is always zero.

Consider the prototype LP described earlier where the rhs value of production constraint is 60. In Fig. 11.10 we show how the feasible region changes and when the set of binding constraints change as we perturb the rhs value of the production constraint. Notice that in Fig. 11.10a the storage constraint drops out of the set of binding constraints and in Fig. 11.10c the demand constraint becomes binding. In between these two extremes, the set of binding constraints, as shown in Fig. 11.10b, remains unchanged. The range over which the current optimal shadow price of 78.57 remains unchanged is from 37.5 to 65.5 (allowable increase is 5.5 and allowable decrease is 22.5). That is, if the rhs of the production constraint were to vary in the range from 37.5 to 65.5 (values of $b_1 \in [37.5, 65.5]$) the shadow price would be constant at 78.57.

Currently, the value of $b_1 = 60$. In Fig. 11.11 we plot the optimal objective value $Z^\star(b_1)$ as a function of b_1, the rhs of production constraint , when b_1 is in the range [37.5, 65.5]. All other values are kept the same. Notice, as we vary b_1, the optimal objective value changes linearly at the rate of the shadow price, i.e., 78.57.

When the reduced cost of a decision variable is nonzero (implying that the value of that decision variable is zero in the optimal solution), the reduced cost is also reflected in the allowable range of its objective coefficient. In this case, one of the allowable limits is always infinite (because making the objective coefficient less attractive will never cause the optimal solution to include the decision variable in the optimal solution); and the other limit, by definition, is the reduced cost (for it is the amount by which the objective coefficient must improve before the optimal solution changes).

2.5.1 One Hundred Percent Rule

While performing sensitivity analysis changes to the objective coefficients, rhs values, or consumption levels are analyzed one at a time. Changing these objective coefficients or rhs values simultaneously does not guarantee that the optimal

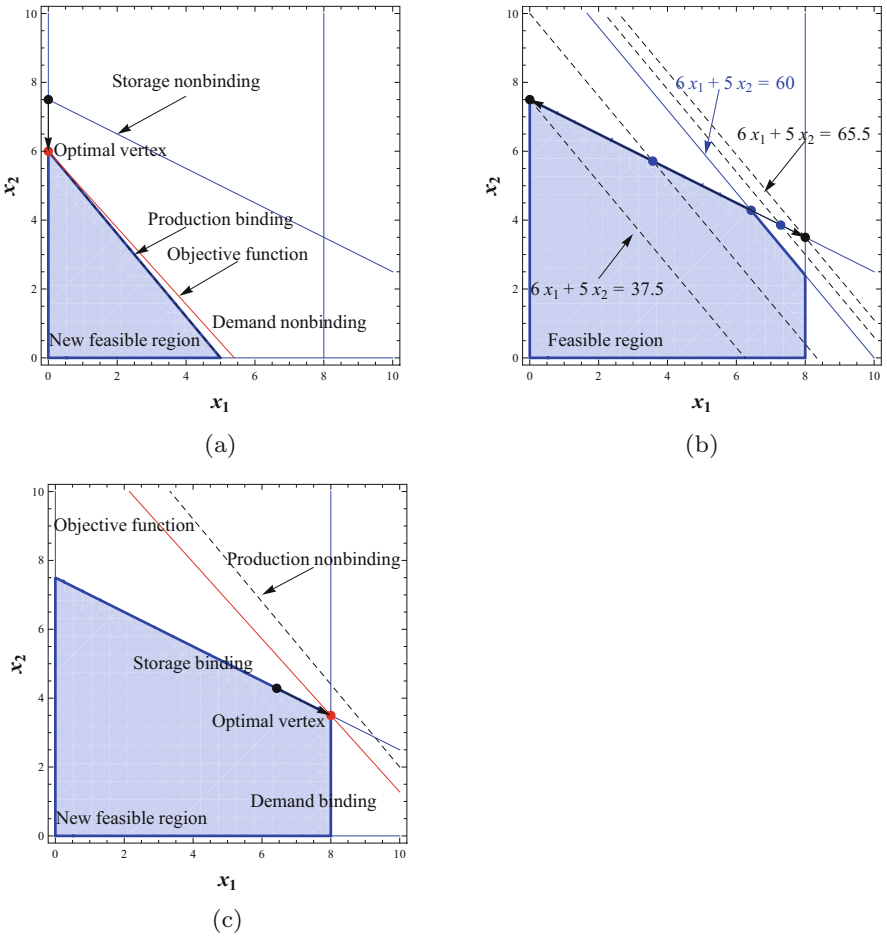


Fig. 11.10 A shadow price is valid until the set of binding constraints remains the same. (**a**) Decreasing the rhs beyond the range. (**b**) The range of the rhs for which shadow price remains constant. (**c**) Increasing the rhs beyond the range

solution is conserved. Simultaneous changes can be implemented and a conservative bound on these simultaneous changes can be computed using the **100% rule**. First, we compute the ranges for the rhs values assuming changes are made one at a time. The 100% rule implies the following: If simultaneous changes are made to the rhs (or the objective coefficients) values of more than one constraint (variable) in such a way that the sum of the fractions of allowable range utilized by these changes is less than or equal to one, the optimal basis (variables that are included in the optimal decision) remains unchanged. Consider the example described earlier where the rhs value of constraint C1 is 60. If we solve the LP, it turns out that at optimality the allowable range for the shadow price of 78.57, in the current optimal solution, is 37.5–65.5. That is, if the rhs of constraint C1 were to be in the range 37.5–65.5

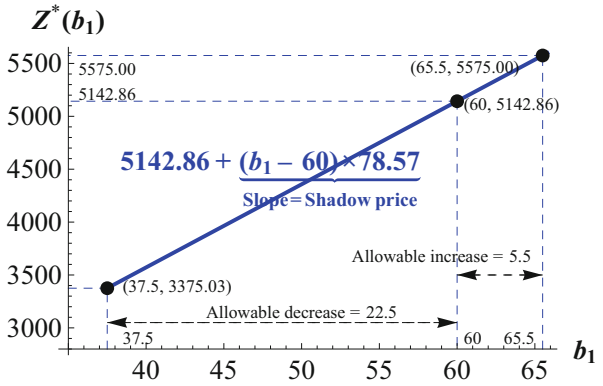


Fig. 11.11 Plot of $Z^{\star}(b_1)$ vs. b_1 for the production constraint, when $b_1 \in [37.5,\ 65.5]$

(values of $b_1 \in [37.5,\ 65.5]$), the shadow price of C1 at optimality would be 78.57 for all these values of the rhs. Further, notice that the current rhs value of constraint C2 is 150 and the range for its shadow price (2.86, corresponding to the current optimal solution) is between 128 and 240. Suppose we reduce the rhs value of the first constraint to $b_1^{\text{new}} \leq 60$ and increase the rhs value of the second constraint to $b_2^{\text{new}} \geq 150$. The 100% rule suggests that the current solution (basis) remains optimal as long as

$$\frac{60 - b_1^{\text{new}}}{60 - 37.5} + \frac{b_2^{\text{new}} - 150}{240 - 150} \leq 1.$$

2.6 A Quick Note About LP Optimality

It is evident from the earlier discussion that any optimal solution to a LPP has a very specific structure. We reiterate the optimal structure of any LPP below:

1. The shadow price of nonbinding constraint is always 0. A binding constraint may have a nonzero shadow price. Together, this implies

   Slack (or surplus) on a constraint × shadow price of the constraint = 0.

2. Every decision variable has a reduced cost associated with it. Basic variables, at optimality, have a zero reduced cost and nonbasic variables may have a nonzero reduced cost. This implies that

   Reduced cost of a variable × the optimal value of the variable = 0.

3. Finally, it is easy to verify that for a LP, at optimality

$$\begin{aligned}\text{The optimal objective value} &= \text{Product of the rhs value of a constraint}\\ &\quad \times \text{the shadow price of the constraint,}\\ &\quad \text{summed over all the constraints, i.e.,}\end{aligned}$$

$$\sum_{j=1}^{n} c_j x_j^\star = \sum_{i=1}^{m} b_i y_i^\star,$$

where $y_i^\star$ is the shadow price of the ith constraint at optimality, b_i is the value of the rhs of constraint i, c_j is the objective coefficient of the jth decision variable, and $x_j^\star$ is the optimal value of the jth decision variable. For the prototype problem described earlier,

$$\sum_{j=1}^{n} c_j x_j^\star = (500 \times 6.429) + (450 \times 4.285) = 5142.8.$$

$$\sum_{i=1}^{m} b_i y_i^\star = (60 \times 78.571) + (150 \times 2.857) = 5142.8.$$

Conditions (1) and (2) together are called the **complementary slackness conditions of optimality**. All the three conditions, (1), (2), and (3) provide an easily **verifiable certificate of optimality** for any LPP. This is one of the fascinating features of any LP optimal solution—the certificate of optimality comes with the solution. Thus, combining the search from vertex to vertex and examining the solution for optimality gives an algorithm (the Simplex algorithm) to solve LPs very efficiently!

3 Methods in Optimization: Integer Programming—Enforcing Integrality Restrictions on Decision Variables

Introducing integrality constraints on decision variables has its advantages. First, we can model more realistic business requirements (economic indivisibility, for example) and second, the model allows us more flexibility such as modeling business logic using binary variables. However, the main disadvantage lies in the

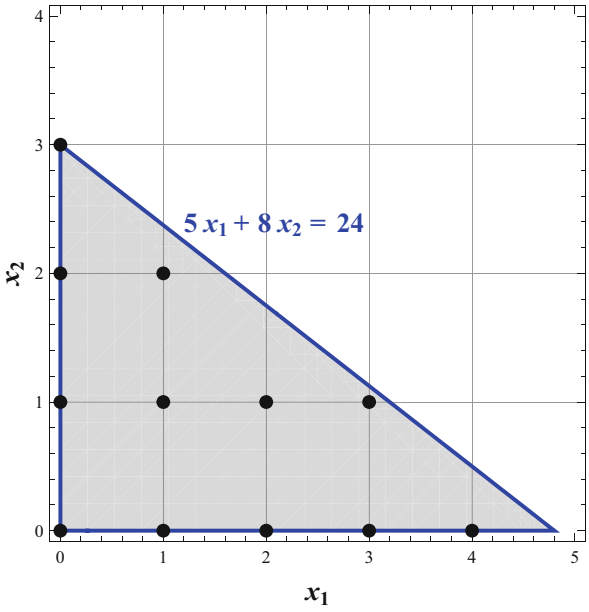


Fig. 11.12 Integer feasible region for (P)

difficulty of solving the model and guaranteeing optimality of the solution. Let us consider a simple example to understand where the computational challenge arises. Consider the following example:

$$\begin{aligned} \text{(P)} \quad \max \quad & 3x_1 + 4x_2 \\ \text{s.t.} \quad & 5x_1 + 8x_2 \leq 24, \\ & x_1,\ x_2 \geq 0 \quad \text{and integer.} \end{aligned}$$

What is the optimal solution for this problem?

Notice that the mathematical representation is very similar to the corresponding LP with the added constraint that both x_1 and x_2 must be restricted to integral values. In Fig. 11.12 we represent the feasible region of this problem.

It is noteworthy that the LP relaxation, i.e., when we ignore the integrality restriction on both the decision variables, is the entire gray region included in the triangle. However, the integer formulations must restrict any solution to the lattice points within the LP feasible region. This "smallest" polyhedral set including all the lattice points is sometimes referred to as the **convex hull** of the integer programming problem. It is readily observable that the LP relaxation includes the integer programming problem's (IP) feasible region (**convex hull**) as a strict subset. One may argue that it should be possible to solve the LP relaxation of (P) and then simply round up or round down the optimal decision variables appropriately.

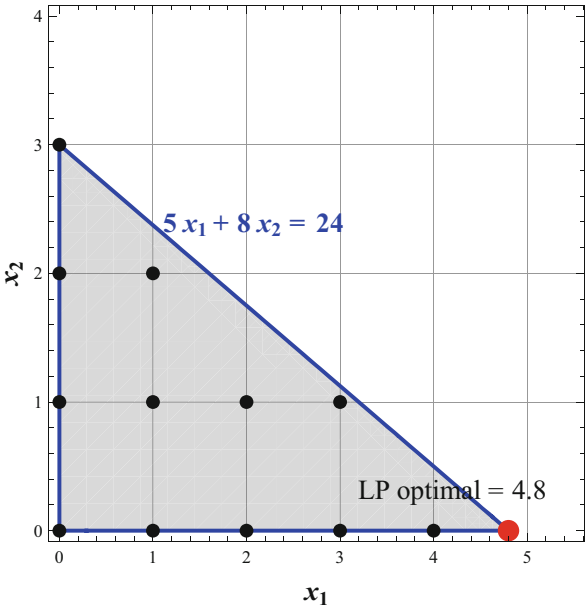


Fig. 11.13 LP solution to (P)—it's not integral valued

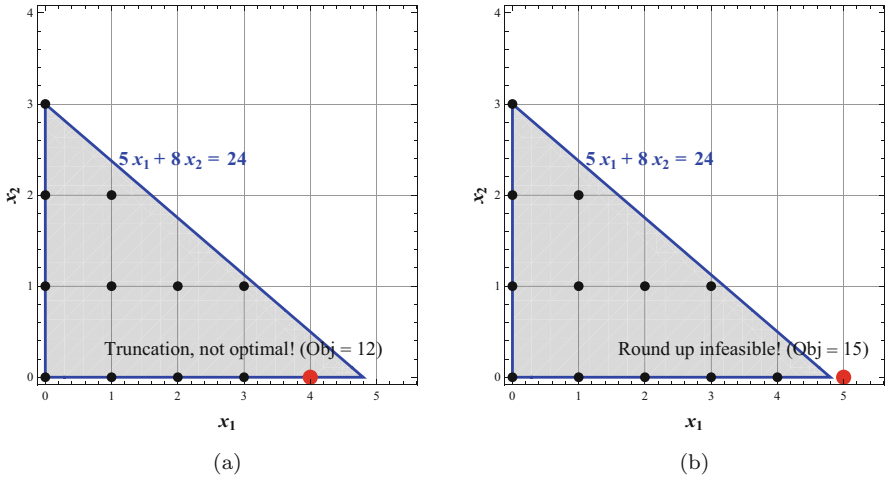


Fig. 11.14 Finding an integer solution by truncating or rounding-up a LP solution may not work. (**a**) Truncating (not optimal). (**b**) Rounding-up (infeasible)

But as Fig. 11.13 illustrates the corner point, at which the LP will always find its optimal solution, need not be integer valued. In this examples the LP relaxation optimal value is found at the vertex (4.8,0). As Fig. 11.14 shows, truncating or rounding-up the optimal LP solution doesn't provide the integer optimal solution.

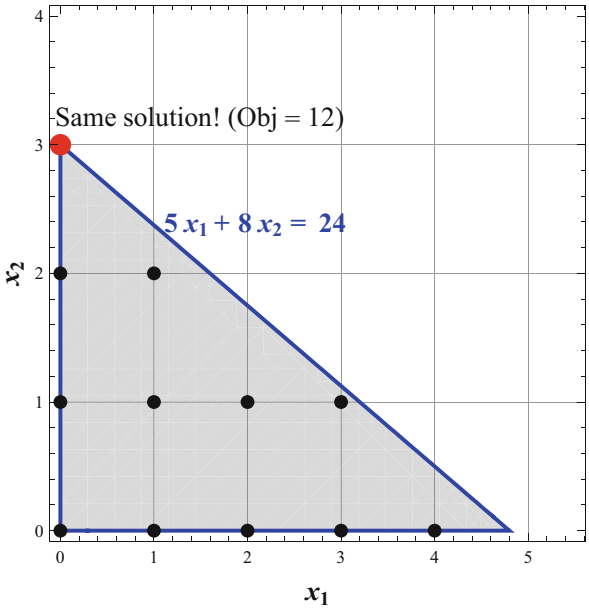


Fig. 11.15 Had the truncated solution been optimal, the LP would have found it at another corner point! That's why it is not optimal

While rounding-up renders the solution infeasible, had the truncated solution been optimal, the LP would have found at another corner point as shown in Fig. 11.15.

In this simple example, it turns out that the IP optimal solution is in the interior of the LP feasible region as shown in Fig. 11.16.

Thus, finding the IP optimal solution is much harder than looking for optimal solution of the LP relaxation (which is guaranteed to be found at a corner point of the LP polyhedra if an optimal solution exists) because the solution can lie in the interior of the corresponding LP feasible region. If it were possible to get an accurate mathematical (polyhedral) description of the **convex hull** using linear constraints, then one could solve the resulting problem (after including these additional constraints) as a LP and guarantee that the LP corner point solution would indeed be optimal to the IP too. However, there is no known standard technique to develop these constraints systematically for any IP and get an accurate mathematical description of the convex hull. Developing such constraints are largely problem specific and tend to exploit the specific mathematical structure underlying the formulation.

So what carries over from LPs to IPs (or MILPs)? The idea of feasibility is unchanged. One can define and compute shadow price in an analogous fashion. The linear relaxation to an integer problem provides a bound on the attainable solution but does not say anything about the feasibility of the problem. There is no way to verify optimality from the solution—instead one must rely on other methods to

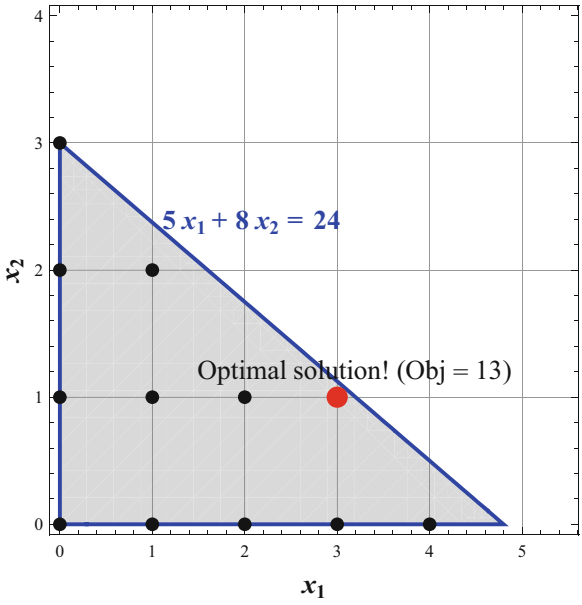


Fig. 11.16 Unfortunately, the IP optimal solution is not at a "corner" point of the original LP feasible region. It is much harder to find

verify optimality. Search methods are used but not going from vertex to vertex! That's why IPs are so hard to solve. Of course, this is not to imply that problems shouldn't be modeled and solved as IPs. Today's state-of-the-art algorithms are very efficient in solving large instances of IPs to optimality but, unlike the case of LPs, guaranteeing optimality is not possible in general. A detailed study of the theory and practice of integer programming can be found in Bertsimas and Tsitsiklis (1997), Nemhauser and Wolsey (1988), and Schrijver (1998).

Next we briefly illustrate a basic **branch-and-bound** solution technique to solve IPs.

3.1 The Branch-and-Bound Method

The basic idea behind the naive branch-and-bound (B&B) method is that of *divide and conquer*. Notice that the feasible region of the LP relaxation of an IP, i.e., when we ignore the integrality constraints, is always larger than that of the feasible region of the IP. Consequently, any optimal solution to the LP relaxation provides a bound on the optimal IP value. In particular, for a minimization problem the LP relaxation will result in a lower bound and for a maximization problem it will result in a upper bound. If $Z^{\star}_{LP}$ denotes the optimal objective value of the LP relaxation and $Z^{\star}_{IP}$ denotes the optimal solution to the IP, then

$$Z^{\star}_{LP} \geq Z^{\star}_{IP} \qquad \text{for a maximization problem, and}$$

$$Z^{\star}_{LP} \leq Z^{\star}_{IP} \qquad \text{for a minimization problem.}$$

The B&B method divides the feasible region (partitions it) and solves for the optimal solution over each partition separately. Suppose F is the feasible region of the IP and we wish to solve $\min_{x \in F} \mathbf{c}'\mathbf{x}$. Consider a partition $F_1, \ldots F_k$ of F. Recollect, a partition implies that the subsets are collectively exhaustive and mutually exclusive, i.e.,

$$F_i \bigcap F_j = \emptyset \quad \text{and} \quad \bigcup_{i=1}^{k} F_i = F.$$

Then, for a minimization problem (equivalently for a maximization problem),

$$\min_{x \in F} \mathbf{c}'\mathbf{x} = \min_{1 \leq i \leq k} \left\{ \min_{x \in F_i} \mathbf{c}'\mathbf{x} \right\}.$$

In other words, we optimize over each subset separately. The idea hinges on the fact that if we can't solve the original problem directly, we might be able to solve the smaller subproblems recursively. Dividing the original problem into subproblems is the idea of **branching**. As is readily observable, a naive implementation of the B&B is equivalent to complete enumeration and can take a arbitrarily long time to solve.

To reduce the computational time most B&B procedures employ an idea called **pruning**. Suppose we assume that each of our decision variables have finite upper and lower bounds (not an unreasonable assumption for most business problems). Then, any feasible solution to our minimization problem provides an upper bound $u(F)$ on the optimal IP objective value.[3] Now, after branching, we obtain a lower bound $b\,(F_i)$ on the optimal solution for each of the subproblems. If $b\,(F_i) \geq u\,(F)$, then we don't need to consider solving the subproblem i any further. This is because we already have a solution better than any that can be found in partition F_i. One typical way to find the lower bound $b\,(F_i)$ is by solving the LP relaxation. Eliminating exploring solution in a partition by creating an appropriate bound is called **pruning**. The process of iteratively finding better values of $b\,(F_i)$ and $u\,(F)$ is called **bounding**. Thus, the basic steps in a **LP-based B&B procedure** involve:

LP relaxation: first solve the LP relaxation of the original IP problem. The result is one of the following:

1. The LP is infeasible $\Longrightarrow$ IP is infeasible.
2. The LP is feasible with an integer solution $\Longrightarrow$ Optimal solution to the IP.
3. LP is feasible but has a fraction solution $\Longrightarrow$ Lower bound for the IP.

In the first two cases of step 1, we are done. In the third case, we must branch and recursively solve the resulting subproblems.

Branching: The most common way to branch is as follows: Select a variable i whose value x_i is fractional in the LP solution. Create two subproblems: in

[3] Typically, we could employ a heuristic procedure to obtain an upper bound to our problem.

one subproblem, impose the constraint $x_i \geq \lceil x_i \rceil$. In the other subproblem, impose the constraint $x_i \leq \lfloor x_i \rfloor$. This is called a **branching rule** (it is the simplest branching rule). Notice that doing so creates two subproblems yet does not eliminate any integer feasible solutions to the original problem. Hence, this branching rule is **valid**, i.e., the constraints generated are **valid inequalities**.

Pruning: After branching we solve the subproblems recursively. Now we consider the following: if the optimal objective value of the LP relaxation is greater than the current upper bound, we need not consider the current subproblem further (pruning), that is, if $Z_{LP}^{i\star} > Z_{IP}$, then prune subproblem i. This is the key to the potential efficiency of the problem.

Before we summarize the steps of the B&B algorithm, we describe some implementation terminology. If we picture the subproblems graphically, they form a **search tree**. Each subproblem is linked to its **parent** and eventually to its **children**. Eliminating a problem from further consideration is called pruning. The act of bounding and then branching is called **processing**. A subproblem that has not yet been considered is called a **candidate** for processing. The set of candidates for processing is called the **candidate list**. Using this terminology, the LP-based B&B procedure (for a minimization problem) can be summarized as follows:

1. To begin, we find an upper bound U using a preprocessing/heuristic routine.
2. We start with the original problem on the candidate list.
3. Select problem S from the candidate list and solve the LP relaxation to obtain the lower bound $b(S)$.
   (a) If LP is infeasible $\Longrightarrow$ candidate is pruned.
   (b) Otherwise, if $b(S) \geq U \Longrightarrow$ candidate is pruned.
   (c) Otherwise, if $b(S) < U$ and the solution is feasible for the IP $\Longrightarrow$ update $U \leftarrow b(S)$.
   (d) Otherwise, *branch* and add the new subproblem to the candidate list.
4. If the candidate list is nonempty, go to step 2. Otherwise, the algorithm is done.

There are several ways to select a candidate in step 2. The **best-first** technique chooses a candidate with the lowest lower bound. Other possibilities are to use a **depth-first** or **breadth-first** technique. The depth-first technique is most common. The depth-first and breadth-first techniques differ in the way the B&B search tree is traversed to select the next candidate to the explored. The reader is referred to Nemhauser and Wolsey (1988) for details on the search procedures. Detail of the B&B algorithm and other procedures such as the cutting plane algorithm can also be found in Wolsey (1998). It is noteworthy that most commercial solvers build on the basic B&B procedure described here and combine it with generating constraints automatically, reducing the number of binary/integer variables used, and using pre- and postprocessing heuristics to generate "good" (from a business implementation standpoint) and feasible IP solutions quickly. For a partial list of LP and IP solvers available to solve large IPs, the reader is referred to https://neos-server.org/neos/solvers/index.html (accessed on Jul 22, 2018).

3.2 A B&B Solution Tree Example

To illustrate the implementation of the LP-based B&B algorithm, consider the following binary variable problem:

$$\begin{aligned} \text{(P1)}\max \quad & 8x_1 + 11x_2 + 6x_3 + 4x_4, \\ \text{s.t.} \quad & 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14, \\ & x_i \in \{0, 1\} \quad \forall\, i = 1, \ldots, 4. \end{aligned}$$

The linear relaxation of optimal solution is $\mathbf{x}^\star = \{1, 1, , 0.5, 0\}$ with an objective value of 22. Notice that this solution is not integral. So we choose a fractional variable to branch on, which is x_3. Essentially, we generate two subproblems: one with the constraints $x_3 = 0$ and the other with the constraint $x_3 = 1$. We illustrate the entire B&B solution tree in Fig. 11.17.

The solution tree shows the LP-relaxation upper bounds (since this is maximization problem) at each node (subproblem) and the variables that were branched on at each iteration (these are the fractional valued variables in the LP solution at that node). The integer valued solutions are marked in red, which provide the lower bounds. We employ the depth-first search process to select candidates to solve

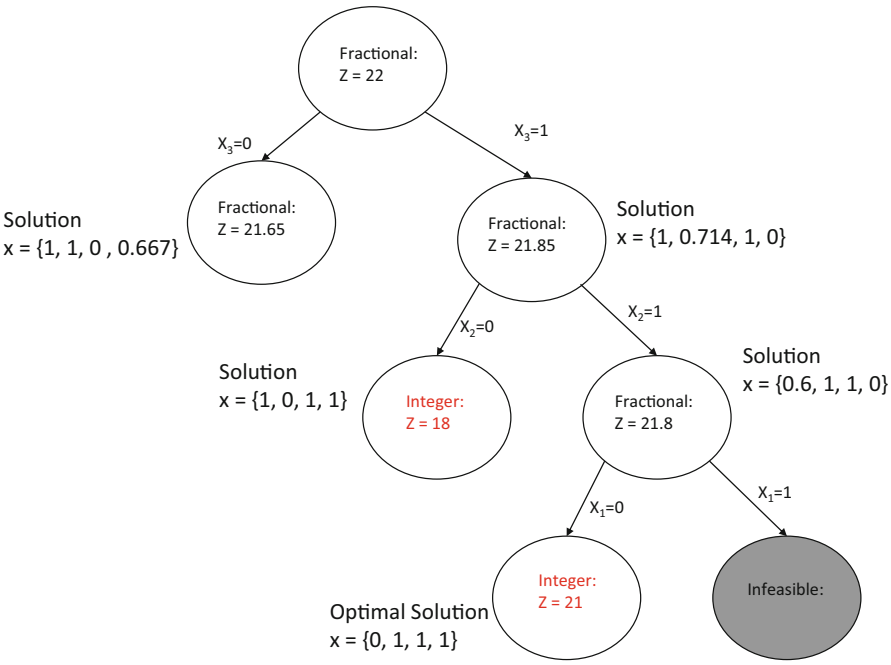


Fig. 11.17 LP-based B&B solution tree for problem P1

iteratively and always choose the "right-side" child candidate. Thus, we begin by branching on x_3 followed by x_2 and then x_1. Notice that by fixing $x_1 = x_2 = x_3 = 1$ we arrive at an infeasible solution at the rightmost node of the tree (fourth level). However, the left child candidate at the same level, i.e., when $x_3 = x_2 = 1$ and $x_1 = 0$, gives us an integer feasible solution with objective value $Z_{IP} = 21$. This is the best IP lower bound solution we have so far—our incumbent IP solution (0,1,1). Now, when we step one level higher to explore the node when $x_3 = 1$ and $x_2 = 0$, we get a LP solution with an objective value 18 (also happens to be integer valued), which is lesser than 21, our incumbent IP solution. Hence, we prune the sub-tree (not shown in the figure) rooted at that node (where the optimal objective value is 18). Similarly, we don't need to explore the sub-tree to the left of the root node, i.e., when we fix $x_3 = 0$ because that sub-tree can never get us a better integer solution than what we already have with our incumbent solution.

4 Methods in Optimization: Nonlinear Optimization Models

A vast majority of problems in real business applications are essentially nonlinear in nature. In fact, linear programming models are a subset of nonlinear models. One may also consider LP models to be an approximation of the real problem. In this section, we discuss a few examples of nonlinear optimization models in statistics, econometrics, and data analytics. However, we do not discuss the algorithmic details of the nonlinear solution techniques. For a detailed discussion on the theory and application of nonlinear programming we refer the readers to Bazaraa et al. (2013), Bertsekas (1999), and Boyd and Vandenberghe (2004). We begin by illustrating the use of optimization in simple linear regression.

4.1 Estimating Coefficients in an Ordinary Least Squares Regression

Optimization plays a very important role in the estimation of coefficients in linear regression models. Consider the method of ordinary least squares (OLS) where, using sample data, we wish to estimate the parameters of a linear relationship between the dependent (**response**) variable $\mathbf{Y} = \langle Y_1, \ldots, Y_n \rangle$ and the corresponding independent (**predictor**) variable $\mathbf{X} = \langle X_1, \ldots, X_n \rangle$, i.e., using the sample observations we try to fit a linear relationship:

$$y_i = \hat{\beta}_1 + \hat{\beta}_2 x_i + \hat{\varepsilon}_i \quad \forall\, i = 1, \ldots, n. \tag{11.14}$$

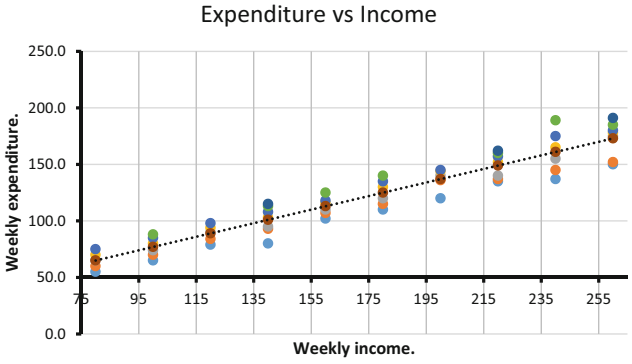
In (11.14), ε_i is the random error (residual) associated with the ith observation and $\left(\hat{\beta}_1, \hat{\beta}_2\right)$ are unbiased estimates of the (true) parameters of the linear function (β_1, β_2). Alternately, the relationship can be expressed as $Y_i = \mathbb{E}[Y \mid x_i] + \varepsilon_i$, where $\mathbb{E}[Y \mid x_i] = \beta_1 + \beta_2 x_i$ is the conditional expectation of all the responses, Y_i, observed when the predictor variable takes a value x_i. It is noteworthy that capital letters indicate random variables and small letters indicate specific values (instances). For example, suppose we are interested in computing the parameters of a linear relationship between a family's weekly income level and its weekly expenditure. In this case, the weekly income level is the predictor (x_i) and the weekly expense is the response (y_i). Figure 11.18a shows a sample of such data collected, i.e., sample of weekly expenses at various income levels. The scatterplot in Fig. 11.18b shows the fitted OLS regression line.

In order to construct the unbiased estimates $\left(\hat{\beta}_1, \hat{\beta}_2\right)$ OLS involves minimizing the sum of squared errors, i.e.,

$$\min \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} \left(y_i - \hat{\beta}_1 - \hat{\beta}_2 x_i\right)^2. \tag{11.15}$$

	X (Income level)									
	80	100	120	140	160	180	200	220	240	260
Y (Expense)	55	65	79	80	102	110	120	135	137	150
	60	70	84	93	107	115	136	137	145	152
	65	74	90	95	110	120	140	140	155	175
	70	80	94	103	116	130	144	152	165	178
	75	85	98	108	118	135	145	157	175	180
		88		113	125	140		160	189	185
				115				162		191
Total	325	462	445	707	678	750	685	1043	966	1211
E[Y|X]	65	77	89	101	113	125	137	149	161	173

(a)



(b)

Fig. 11.18 Fitting an OLS regression line. (**a**) Weekly expenditure at various income levels. (**b**) Weekly expenditure as a function of weekly expense

For a give sample, notice that $\left[\sum_{i=1}^{n} y_i\right]$, $\left[\sum_{i=1}^{n} y_i x_i\right]$, $\left[\sum_{i=1}^{n} x_i^2\right]$, $\left[\sum_{i=1}^{n} y_i^2\right]$ are constants. Hence, (11.15) is a simple quadratic function of the parameters, i.e.,

$$-2\hat{\beta}_1 \left[\sum_{i=1}^{n} y_i\right] - 2\hat{\beta}_2 \left[\sum_{i=1}^{n} y_i x_i\right] + \left(\hat{\beta}_2\right)^2 \left[\sum_{i=1}^{n} x_i^2\right] + \left[\sum_{i=1}^{n} y_i^2\right]$$

and represents an unconstrained quadratic optimization problem. There are two ways of solving this problem. One, we can use differential calculus and solve it by setting derivatives to equal zero. We get what are known as normal equations (see Chap. 7 on linear regression). Gujarati (2009) also provides a detailed description of the analytical solution to this nonlinear parameter estimation optimization problem. Two, we can use a descent method as follows:

Step 1: We start with an initial solution (may be computed using a heuristic approach).

Step 2: We then find a value improving direction and move along that direction by changing $\hat{\beta}_1$ and $\hat{\beta}_2$, slightly.

Step 3: Repeat the steps 1 and 2, until the gain from such a move is very small (**stopping criteria**).

Like in LPPs, this problem does not have local optimal solutions—in other words, once we are unable to improve the solution we know we are at or close to the global optimal solution.

4.2 Estimating Coefficients Using Maximum Likelihood Estimation

As described in Sect. 4.1 we consider a two-variable model:

$$y_i = \hat{\beta}_1 + \hat{\beta}_2 x_i + \varepsilon_i$$

where Y_i is the response variable and x_i is the predictor variable. The method of maximum likelihood estimation (MLE), like the OLS method, helps us estimate the linear regression parameters $\left(\hat{\beta}_1,\ \hat{\beta}_2\right)$. In the MLE approach, we assume that the sample collected is made of independent and identically distributed observations $(y_i,\ x_i)$ and that the error terms follow a normal distribution with mean zero and variance σ^2. This implies that Y_i are normally distributed with mean $\beta_1 + \beta_2 x_i$ and variance σ^2. Consequently, the joint probability density function of $Y_1, \ldots,\ Y_n$ can be written as

$$f\left(Y_1, \ldots,\ Y_n \mid \beta_1 + \beta_2 x_i,\ \sigma^2\right).$$

But given that the sample points are drawn independently, we express the joint probability density function as a product of the individual density functions as

$$f\left(Y_1, \ldots, Y_n \mid \beta_1 + \beta_2 x_i, \sigma^2\right) = f\left(Y_1 \mid \beta_1 + \beta_2 x_i, \sigma^2\right) f\left(Y_2 \mid \beta_1 + \beta_2 x_i, \sigma^2\right) \cdots f\left(Y_n \mid \beta_1 + \beta_2 x_i, \sigma^2\right)$$

where

$$f\left(Y_i\right) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left[-\frac{1}{2}\frac{(Y_i-\beta_1-\beta_2 x_i)^2}{\sigma^2}\right]}$$

which is the density function of a normally distributed random variable. For given values of the response variable the likelihood function, $\mathrm{LF}\left(\hat{\beta}_1, \hat{\beta}_2, \sigma^2\right)$, is written as

$$\mathrm{LF}\left(\hat{\beta}_1, \hat{\beta}_2, \sigma^2\right) = \frac{1}{\sigma^n\left(\sqrt{2\pi}\right)^n} e^{\left[-\frac{1}{2}\sum_{i=1}^{n}\frac{\left(Y_i-\hat{\beta}_1-\hat{\beta}_2 x_i\right)^2}{\sigma^2}\right]}.$$

The method of MLE computes $\left(\hat{\beta}_1, \hat{\beta}_2\right)$ such that the probability of observing the given $\mathbf{y} = \langle \mathrm{y}_1, \ldots, \mathrm{y}_n \rangle$ is maximum (as high as possible.) Notice that this is a nonlinear optimization problem that maximizes the likelihood function over $\hat{\beta}_1$ and $\hat{\beta}_2$. One natural way to solve this problem is to convert LF function into its log form, i.e.,

$$\ln \mathrm{LF}\left(\hat{\beta}_1, \hat{\beta}_2, \sigma^2\right) = -n\ln\sigma - \frac{n}{2}\ln(2\pi) - \frac{1}{2}\sum_{i=1}^{n}\frac{\left(Y_i - \hat{\beta}_1 - \hat{\beta}_2 x_i\right)^2}{\sigma^2},$$

$$= -\frac{n}{2}\ln\sigma^2 - \frac{n}{2}\ln(2\pi) - \frac{1}{2}\sum_{i=1}^{n}\frac{\left(Y_i - \hat{\beta}_1 - \hat{\beta}_2 x_i\right)^2}{\sigma^2}.$$

Maximizing the log-likelihood function is a simple unconstrained quadratic optimization problem (just as seen in Sect. 4.1). Once again we refer the readers to Gujarati (2009) for details about the analytical solution to this optimization problem. MLE is also covered in detail in Chap. 8 (advanced regression and missing data) of this book.

4.3 The Logit Model for Classification

Unlike the case discussed in Sect. 4.1, sometimes we encounter situations wherein the response variables take binary outcomes. For example, consider a binary model, in which x_i (predictor) is the price of a product and y_i (response) is whether a customer purchased a product. In this case, the response variable $y_i \in \{0,\ 1\}$. Fitting an OLS regression model, in this case, may not be appropriate because the response variable must be restricted to the interval [0, 1] and there exists no such restriction in the standard linear regression model. Instead, we use a binary outcome model that tries to estimate the conditional probability that $y_i = 1$ as a function of the independent variable, i.e., $\Pr\{Y_i = 1 \mid x_i\} = F\left(\hat{\beta}_1 + \hat{\beta}_2 x_i\right)$, where the function $F(\cdot)$ represents the cumulative density function of a probability distribution. One common model used is the **logit** model**,** where $F(\cdot)$ is the logistic distribution function, i.e.,

$$F\left(\hat{\beta}_1 + \hat{\beta}_2 x_i\right) = \frac{e^{\left[\hat{\beta}_1 + \hat{\beta}_2 x_i\right]}}{1 + e^{\left[\hat{\beta}_1 + \hat{\beta}_2 x_i\right]}}.$$

Assuming that the observations in the sample data are independent of each other, the conditional likelihood of seeing the n outcomes in our sample data is given by

$$\prod_{i=1}^{n} \Pr\{Y = y_i \mid x_i\} = \prod_{i=1}^{n} F\left(\hat{\beta}_1 + \hat{\beta}_2 x_i\right)^{y_i} \times \left[1 - F\left(\hat{\beta}_1 + \hat{\beta}_2 x_i\right)\right]^{(1-y_i)}$$

because $y_i \in \{0,\ 1\}$ and $\langle y_1, \ldots,\ y_n \rangle$ represents a sequence of Bernoulli trials. As described in Sect. 4.2, a natural way to solve this problem is to convert the likelihood function into its log form, i.e.,

$$\begin{aligned}
\ln \text{LF}\left(\hat{\beta}_1, \hat{\beta}_2\right) &= \sum_{i=1}^{n} y_i \ln F\left(\hat{\beta}_1 + \hat{\beta}_2 x_i\right) + \sum_{i=1}^{n} (1 - y_i) \ln\left[1 - F\left(\hat{\beta}_1 + \hat{\beta}_2 x_i\right)\right], \\
&= \sum_{i=1}^{n} y_i \ln\left[\frac{e^{\left[\hat{\beta}_1 + \hat{\beta}_2 x_i\right]}}{1 + e^{\left[\hat{\beta}_1 + \hat{\beta}_2 x_i\right]}}\right] + \sum_{i=1}^{n} (1 - y_i) \ln\left[\frac{1}{1 + e^{\left[\hat{\beta}_1 + \hat{\beta}_2 x_i\right]}}\right], \\
&= \sum_{i=1}^{n} y_i \left[\hat{\beta}_1 + \hat{\beta}_2 x_i - \ln\left(1 + e^{\left[\hat{\beta}_1 + \hat{\beta}_2 x_i\right]}\right)\right] \\
&\quad - \sum_{i=1}^{n} (1 - y_i) \left(\ln\left(1 + e^{\left[\hat{\beta}_1 + \hat{\beta}_2 x_i\right]}\right)\right), \\
&= \sum_{i=1}^{n} y_i \left[\hat{\beta}_1 + \hat{\beta}_2 x_i\right] - \sum_{i=1}^{n} \ln\left(1 + e^{\left[\hat{\beta}_1 + \hat{\beta}_2 x_i\right]}\right).
\end{aligned}$$

The optimization problem reduces to choosing the parameters (coefficients) $\hat{\beta}_1$ and $\hat{\beta}_2$ to maximize the log-likelihood function, $\ln \text{LF}\left(\hat{\beta}_1, \hat{\beta}_2\right)$. This is a nonlinear optimization problem but cannot be solved analytically using standard differential calculus. We may have to resort to approximately solving it numerically (e.g., see Newton's method in Bazaraa et al., 2013). It is noteworthy that this type of formulation can be used for making **multi-class predictions/classifications**, where Y can take on more than two values (not just binary). See Chaps. 15, 16, and 17 on machine learning techniques for discussion on these types of problems. Several specialized algorithms have been developed to solve this problem efficiently. Moreover, it is somewhat straightforward to connect this to a machine learning problem! The multi-class prediction can be seen to be equivalent to a single-layer neural network using softmax loss function (see Chaps. 16 and 17 on Supervised Learning and Deep Learning). The connection between learning and optimization is an advanced topic well worth pursuing.

As described in this section, the techniques and solution methodologies for solving nonlinear optimization problems can be varied. For a partial list of algorithmic procedures to solve nonlinear problems the reader is referred to https://neos-guide.org/algorithms (accessed on Jul 22, 2018).

5 Discussion

In this chapter, we touched upon the basics of optimization. In particular, we focused on formulating, solving, and interpreting solutions of LPPs. LPs have been used in a large number of business and scientific applications over the last few decades. It is important to understand that while the LP methodology is very efficient and easy to model, there are larger classes of optimization techniques that help model business and scientific applications even more closer to reality, integer programming being one of them. Finally, we briefly described nonlinear optimization models and showed a few examples that are closely related to often-used econometric models.

For a complete taxonomy of the types of mathematical programs/optimization techniques encountered in theory and practice, we refer the readers to NEOS guide.[4] With data sciences, machine learning, and analytics gaining importance, the use of LPs (and optimization methods in general) will continue to grow. In a sense, optimization models will eventually become ubiquitous.

[4] https://neos-guide.org/content/optimization-taxonomy (accessed on Jul 22, 2018).

Appendix

Spreadsheet Models and Excel Solver Reports

There are a few online tutorials available to understand how to input a LP model in Solver. The two commonly used websites are Solver[5] and Microsoft support[6] page. This section describes the various fields in the LP reports generated by Microsoft Solver and how to locate the information related to shadow prices, reduced costs, and their ranges after the model has been solved. We use the prototype example referred earlier to describe these reports.

The Answer Report

Figure 11.19 shows the **answer** report generated by Excel Solver for our prototype problem. We describe the entries in this report.

Target Cell The initial value of the objective function (to be maximized or minimized), and its final optimal objective value.

Adjustable Cells The initial and final values of the decision variables.

Constraints Maximum or minimum requirements that must be met, whether they are met just barely (binding) or easily (not binding), and the values of the slacks (excesses) leftover. Binding constraints have zero slacks and nonbinding ones have positive slacks.

Target Cell (Max)

Cell	Name	Original Value	Final Value
$K$34	PROFIT Z	0	5142.86

Adjustable Cells

Cell	Name	Original Value	Final Value
$G$33	Objective: max 500 x1 + 450 x2 X1	0.00	6.43
$H$33	Objective: max 500 x1 + 450 x2 X2	0.00	4.29

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$I$36	Production	60.000	$I$36<=$K$36	Binding	0
$I$37	Storage	150.000	$I$37<=$K$37	Binding	0
$I$38	Demand	6.429	$I$38<=$K$38	Not Binding	1.571428571

Fig. 11.19 Answer report

[5] https://www.solver.com/excel-solver-online-help (accessed on Jul 22, 2018).

[6] https://support.office.com/en-us/article/define-and-solve-a-problem-by-using-solver-5d1a388f-079d-43ac-a7eb-f63e45925040 (accessed on Jul 22, 2018).

Adjustable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
$G$33	Objective: max 500 x1 + 450 x2 X1	6.43	0.00	500	40	275
$H$33	Objective: max 500 x1 + 450 x2 X2	4.29	0.00	450	550	33.33

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
$I$36	Production	60.000	78.571	60	5.5	22.5
$I$37	Storage	150.000	2.857	150	90	22
$I$38	Demand	6.429	0.000	8	1E+30	1.57

Fig. 11.20 Sensitivity report (shadow prices and validity range)

The Sensitivity Report

Figure 11.20 shows the **sensitivity** report generated by Excel Solver for our prototype problem. Below we describe the entries in this report.

Adjustable Cells The decision variables, their cell addresses, names, and optimal values.

Reduced Cost This relates to decision variables that are bounded, from below (such as by zero in the nonnegativity requirement), or from above (such as by a maximum number of units that can be produced or sold). Recollect:

1. A variable's **reduced cost** is the amount by which the optimal objective value will change if that bound was relaxed or tightened.
2. If the optimal value of the decision variable is at its specified upper bound, the reduced cost is the amount by which optimal objective value will improve (go up in a maximization problem or go down in a minimization problem) if we relaxed the upper bound by increasing it by one unit.
3. If the optimal value of the decision variable is at its lower bound, its reduced cost is the amount by which the optimal objective value will be hurt (go down in a maximization problem or go up in a minimization problem) if we tightened the bound by increasing it by one unit.

Objective Coefficient The unit contribution of the decision variable to the objective function (unit profit or cost).

Allowable Increase and Decrease The amount by which the coefficient of the decision variable in the objective function can change (increase or decrease) before the optimal solution (the values of decision variables) changes. As long as an objective coefficient changes within this range, the current optimal solution (i.e., the values of decision variables) will remain optimal (although the value of the objective function optimal objective value will change as the objective coefficient changes, even within the allowable range).

Shadow Price Recollect:

1. The shadow price associated with each constraints measures the amount of change in the optimal objective value optimal objective value that would result from changing that constraint by a small amount.
2. In general, it is the increase in optimal objective value resulting from an increase in the right-hand side of that constraint.
3. Its absolute value measures the *marginal* (or incremental) *improvement* in optimal objective value (i.e., an increase in the maximum profit or a decrease in the minimum cost) if that constraint was *relaxed* (i.e., if the lower limit was reduced or the upper limit was increased) by one unit. Similarly, it is the *marginal degradation* in optimal objective value (i.e., if the lower limit was raised or the upper limit was reduced) by one unit. For example, if the constraint represents limited availability of a resource, its shadow price is the amount by which the optimal profit will increase if we had a little more of that resource and we used it in the best possible way. It is then the maximum price that we should be willing to pay to have more of this resource. Equivalently, it is the *opportunity cost* of not having more of that resource.

Allowable Increase and Decrease Recollect:

1. This is the amount by which the constraint can be relaxed or tightened before its shadow price changes. if the constraint imposes an upper limit, and it is relaxed by increasing this limit by more than the "allowable increase," the optimal objective value will still improve but at a lower rate, so the shadow price will go down below its current value. Similarly, if the upper limit on the constraint is decreased by more than the "allowable decrease," the optimal objective value will degrade at an even higher rate and its shadow price will go up.
2. If the constraint imposes a lower limit and that constraint is relaxed by decreasing the limit by more than the "allowable decrease," the optimal objective value will still improve but only at a lower rate and the shadow price will decrease. If , on the other hand, the lower limit is increased by more than the "allowable increase," the constraint becomes tighter, the optimal objective value will degrade faster, and the shadow price will increase. Thus, there are decreasing marginal benefits to relaxing a constraint, and increasing marginal costs of tightening a constraint.

It should be noted that all of the information in the sensitivity report assumes that only one parameter is changed at a time. Thus, the effects of relaxing or tightening two constraints or changing the objective coefficients of two decision variables cannot be determined from the sensitivity report. Often, however, if the changes are small enough to be within the respective allowable ranges, the total effect can be determined by simply adding the individual effects.

In an **Excel report** degeneracy can be spotted by looking at the rhs values of any of the constraints. **If the constraints (for the range over which the optimal shadow price is valid) have an allowable increase or allowable decrease of**

zero, then the LP is degenerate. One has to be careful while interpreting optimal solutions for degenerate LPs. For example:

1. When a solution is degenerate, the reduced costs may not be unique. Additionally, the objective function coefficients for the variable cells must change by at least as much (and possibly more than) their respective reduced costs before the optimal solution would change.
2. Shadow prices and their ranges can be interpreted in the usual way, but they are not unique. Different shadow prices and ranges may apply to the problem (even if the optimal solution is unique).

Exercises

Ex. 11.1 (LP Modeling)

Ex. 11.1.1 Retail Outlet Staffing

Consider a retail shop that is open 7 days a week. Based on past experience, the number of workers needed on a particular day is given as follows:

Day	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Number	15	12	17	16	19	14	11

Every employee works five consecutive days and then takes off two days, repeating this pattern indefinitely. Our goal is to minimize the number of employees that staff the outlet. Define your variables, constraints, and objective function clearly.
Develop a Solver model and solve for the optimal staffing plan.

Ex. 11.1.2 Managing a Portfolio

We are going to manage an investment portfolio over a 6-year time horizon. We begin with ₹1,000,000, and at various times we can invest in one or more of the following:

(a) Savings account X, annual yield 6%
(b) Security Y, 2-year maturity, total yield 14% if bought now, 12% thereafter
(c) Security Z, 3-year maturity, total yield 17%
(d) Security W, 4-year maturity, total yield 22%

To keep things simple we will assume that each security can be bought in any denomination. We can make savings deposits or withdrawals anytime. We can buy Security Y any year but year 3. We can buy Security Z anytime after the first year. Security W, now available, is a one-time opportunity. Write down a LP model to maximize the final investment yield. Assume all investments must mature on or before year 6 and you cannot sell securities in between. Define your decision variables and constraints clearly.

Ex. 11.2 (Interpreting the Sensitivity Report)
AC manufactures two television models, Astros and Cosmos. Each Astro set sells for $300 and each Cosmo sells for $250 a set. AC purchases components for an Astro set for $260 and components for a Cosmo set cost $190.

Production of each model involves circuit board fabrication, picture tube construction, and chassis assembly. There are two separate and completely automated lines for circuit board fabrication, one for Astro and one for Cosmo. However, the picture tube and chassis assembly departments are shared in the production of both sets.

The capacity of the Astro circuit board fabrication line is 70 sets per day, while that of the Cosmo line is 50 sets per day. The picture tube department has 20 workstations, while the chassis department has 16 workstations. Each workstation can process one TV set at a time and is operated 6 h a day. Each Astro set takes 1 h for chassis assembly and 1 h for tube production. Each Cosmo set requires 2 h for picture tube production and 1 h for chassis assembly.

Workers in the picture tube and chassis assembly departments are paid $10 an hour. Heating, lighting, and other overhead charges amount to $1000 per day.

1. How many Astros and Cosmos should AC produce each day? What will be the maximum profit?
2. How should they allocate the available resources among the two models? Where are the bottlenecks?
3. Suppose that due to raw material shortage AC could make only 30 circuit boards for Cosmos each day. What will be the effect on their operation?
4. Suppose workers in the picture tube department are willing to work overtime for a premium of $21 an hour. How many hours of overtime, if any, should they employ? How will they use it ?
5. If a workstation in the picture tube department breaks down, how will it affect AC's production and profit ?
6. If a chassis assembly workstation breaks down, how will it affect AC's production plan and profit?
7. How much would you be willing to pay to increase Astro's circuit board capacity?
8. Suppose AC has developed a new model that uses same circuit boards as a Cosmos and requires 3 h of the picture tube time. If its profit margin is expected to be a high $42, should they produce it?
9. If the profit margin on Astro goes up to $40 a set, how would it affect the firm's production plan and the daily profit? What if it goes down by $10 a set?
10. How much must Cosmos's price increase before you will consider producing more Cosmos?

Ex. 11.3 (Modeling with Binary Variables)

1. Consider the knapsack set $X_1 = \{x_1, \} \ldots, x_5 \in \{0, 1\} : 3x_1 - 4x_2 + 2x_3 - 3x_4 + x_5 \leq 2$. Is the constraint $x_2 + x_4 \geq 1$ a valid inequality? Why or why not? *Note*: Suppose we formulate an integer program by specifying a rational

polyhedron $P = \left\{\mathbf{x} \in \mathbb{R}_+^n \mid \mathbf{Ax} \leq \mathbf{b}\right\}$ and define $S = \mathbb{Z}^n \cap P$, where $\mathbb{Z}_+^n$ is the n-dimensional set of nonnegative integers. Thus, $S = \left\{\mathbf{x} \in \mathbb{Z}_+^n \mid \mathbf{Ax} \leq \mathbf{b}\right\}$ and conv(S) is the convex hull of S, i.e., the set of points that are convex combinations of points in S. Note that conv $(S) \subseteq S$; "Ideal" if conv$(S) = S$. An inequality $\pi^T \mathbf{x} \leq \pi_0$ is called a *valid inequality* if it is satisfied by all points in S.

2. Solve using the branch-and-bound (B&B) algorithm. Draw the B&B tree, show your branches, LP solutions, lower and upper bounds. You may simply branch in sequence $x_1$followed by x_2 and so on.

$$\max\ 9x_1 + 3x_2 + 5x_3 + 3x_4$$
$$\text{s.t. } 5x_1 + 2x_2 + 5x_3 + 4x_4 \leq 10$$
$$x_{1,} \ldots, x_4 \in \{0, 1\}.$$

References

Bazaraa, M. S., Jarvis, J. J., & Sherali, H. D. (2011). *Linear programming and network flows*. Hoboken, NJ: Wiley.

Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (2013). *Nonlinear programming: Theory and algorithms*. Hoboken, NJ: Wiley.

Bertsekas, D. P. (1999). *Nonlinear programming*. Belmont, MA: Athena Scientific.

Bertsimas, D., & Tsitsiklis, J. N. (1997). *Introduction to linear optimization* (Vol. 6). Belmont, MA: Athena Scientific.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge University Press.

Bradley, S. P., Hax, A. C., & Magnanti, T. L. (1977). Applied mathematical programming.

Chvátal, V. (1983). *Linear programming*. New York: WH Freeman.

Gujarati, D. N. (2009). *Basic econometrics*. New York: Tata McGraw-Hill Education.

Luenberger, D. G., & Ye, Y. (1984). *Linear and nonlinear programming* (Vol. 2). Berlin: Springer.

Nemhauser, G. L., & Wolsey, L. A. (1988). *Interscience series in discrete mathematics and optimization: Integer and combinatorial optimization*. Hoboken, NJ: Wiley.

Schrijver, A. (1998). *Theory of linear and integer programming*. Chichester: Wiley.

Wagner, H. M. (1969). *Principles of operations research: With applications to managerial decisions*. Upper Saddle River, NJ: Prentice-Hall.

Wolsey, L. A. (1998). *Integer programming*. New York: Wiley.