1
Introduction

OPTIMIZATION IS A TECHNOLOGY that can be used to devise effective *decisions* or *predictions* in a variety of contexts, ranging from production planning to engineering design and finance, to mention just a few. In simplified terms, the process for reaching the decision starts with a phase of construction of a suitable mathematical *model* for a concrete problem, followed by a phase where the model is *solved* by means of suitable numerical algorithms. An optimization model typically requires the specification of a quantitative *objective criterion* of goodness for our decision, which we wish to maximize (or, alternatively, a criterion of cost, which we wish to minimize), as well as the specification of *constraints*, representing the physical limits of our decision actions, budgets on resources, design requirements that need be met, etc. An optimal design is one which gives the best possible objective value, while satisfying all problem constraints.

In this chapter, we provide an overview of the main concepts and building blocks of an optimization problem, along with a brief historical perspective of the field. Many concepts in this chapter are introduced without formal definition; more rigorous formalizations are provided in the subsequent chapters.

1.1 Motivating examples

We next describe a few simple but practical examples where optimization problems arise naturally. Many other more sophisticated examples and applications will be discussed throughout the book.

1.1.1 Oil production management

An oil refinery produces two products: jet fuel and gasoline. The profit for the refinery is \$0.10 per barrel for jet fuel and \$0.20 per barrel for gasoline. Only 10,000 barrels of crude oil are available for processing. In addition, the following conditions must be met.

1. The refinery has a government contract to produce at least 1,000 barrels of jet fuel, and a private contract to produce at least 2,000 barrels of gasoline.

2. Both products are shipped in trucks, and the delivery capacity of the truck fleet is 180,000 barrel-miles.

3. The jet fuel is delivered to an airfield 10 miles from the refinery, while the gasoline is transported 30 miles to the distributor.

How much of each product should be produced for maximum profit?

Let us formalize the problem mathematically. We let x_1, x_2 represent, respectively, the quantity of jet fuel and the quantity of gasoline produced, in barrels. Then, the profit for the refinery is described by function $g_0(x_1, x_2) = 0.1x_1 + 0.2x_2$. Clearly, the refinery interest is to maximize its profit g_0. However, constraints need to be met, which are expressed as

$$\begin{aligned} x_1 + x_2 &\leq 10,000 \quad \text{(limit on available crude barrels)} \\ x_1 &\geq 1,000 \quad \text{(minimum jet fuel)} \\ x_2 &\geq 2,000 \quad \text{(minimum gasoline)} \\ 10x_1 + 30x_2 &\leq 180,000 \quad \text{(fleet capacity).} \end{aligned}$$

Therefore, this production problem can be formulated mathematically as the problem of finding x_1, x_2 such that $g_0(x_1, x_2)$ is maximized, subject to the above constraints.

1.1.2 *Prediction of technology progress*

Table 1.1 reports the number N of transistors in 13 microprocessors as a function of the year of their introduction.

If one observes a plot of the logarithm of N_i versus the year y_i (Figure 1.1), one sees an approximately linear trend. Given these data, we want to determine the "best" line that approximates the data. Such a line quantifies the trend of technology progress, and may be used to estimate the number of transistors in a microchip in the future. To model this problem mathematically, we let the approximating line be described by the equation

$$z = x_1 y + x_2, \tag{1.1}$$

where y is the year, z represents the logarithm of N, and x_1, x_2 are the unknown parameters of the line (x_1 is the slope, and x_2 is the

year: y_i	no. transistors: N_i
1971	2250
1972	2500
1974	5000
1978	29000
1982	120000
1985	275000
1989	1180000
1993	3100000
1997	7500000
1999	24000000
2000	42000000
2002	220000000
2003	410000000

Table 1.1 Number of transistors in a microprocessor at different years.

intercept of the line with the vertical axis). Next, we need to agree on a criterion for measuring the level of *misfit* between the approximating line and the data. A commonly employed criterion is one which measures the sum of squared deviations of the observed data from the line. That is, at a given year y_i, Eq. (1.1) predicts $x_1 y_i + x_2$ transistors, while the observed number of transistors is $z_i = \log N_i$, hence the squared error at year y_i is $(x_1 y_i + x_2 - z_i)^2$, and the accumulated error over the 13 observed years is

$$f_0(x_1, x_2) = \sum_{i=1}^{13} (x_1 y_i + x_2 - z_i)^2.$$

The best approximating line is thus obtained by finding the values of parameters x_1, x_2 that minimize the function f_0.

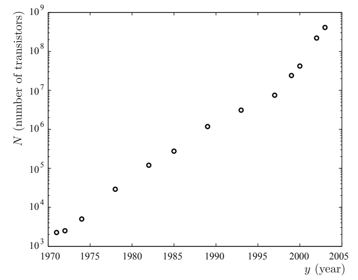

Figure 1.1 Semi-logarithmic plot of the number of transistors in a microprocessor at different years.

1.1.3 *An aggregator-based power distribution model*

In the electricity market, an *aggregator* is a marketer or public agency that combines the loads of multiple end-use customers in facilitating the sale and purchase of electric energy, transmission, and other services on behalf of these customers. In simplified terms, the aggregator buys wholesale c units of power (say, Megawatt) from large power distribution utilities, and resells this power to a group of n business or industrial customers. The i-th customer, $i = 1, \ldots, n$, communicates to the aggregator its ideal level of power supply, say c_i Megawatt. Also, the customer *dislikes* to receive more power than its ideal level (since the excess power has to be paid for), as well as it dislikes to receive less power that its ideal level (since then the customer's business may be jeopardized). Hence, the customer communicates to the aggregator its own model of dissatisfaction, which we assume to be of the following form

$$d_i(x_i) = \alpha_i (x_i - c_i)^2, \quad i = 1, \ldots, n,$$

where x_i is the power allotted by the aggregator to the i-th customer, and $\alpha_i > 0$ is a given, customer-specific, parameter. The aggregator problem is then to find the power allocations x_i, $i = 1, \ldots, n$, so as to minimize the average customer dissatisfaction, while guaranteeing that the whole power c is sold, and that no single customer incurs a level of dissatisfaction greater than a contract level $\bar{d}$.

The aggregator problem is thus to minimize the average level of customer dissatisfaction

$$f_0(x_1, \ldots, x_n) = \frac{1}{n} \sum_{i=1}^{n} d_i(x_i) = \frac{1}{n} \sum_{i=1}^{n} \alpha_i (x_i - c_i)^2,$$

while satisfying the following constraints:

$$\begin{aligned} \sum_{i=1}^{n} x_i = c, &\quad \text{(all aggregator power must be sold)} \\ x_i \geq 0,\ i = 1, \ldots, n, &\quad \text{(supplied power cannot be negative)} \\ \alpha_i (x_i - c_i)^2 \leq \bar{d},\ i = 1, \ldots, n, &\quad \text{(dissatisfaction cannot exceed } \bar{d}\text{).} \end{aligned}$$

1.1.4 *An investment problem*

An investment fund wants to invest (all or in part) a total capital of c dollars among n investment opportunities. The cost for the i-th investment is w_i dollars, and the investor expects a profit p_i from this investment. Further, at most b_i items of cost w_i and profit p_i are available on the market ($b_i \leq c/w_i$). The fund manager wants to know how many items of each type to buy in order to maximize his/her expected profit.

This problem can be modeled by introducing decision variables x_i, $i = 1, \ldots, n$, representing the (integer) number of units of each investment type to be bought. The expected profit is then expressed by the function

$$f_0(x_1, \ldots, x_n) = \sum_{i=1}^{n} p_i x_i.$$

The constraints are instead

$$\begin{aligned} \sum_{i=1}^{n} w_i x_i \leq c, &\quad \text{(limit on capital to be invested)} \\ x_i \in \{0, 1, \ldots, b_i\}, \quad i = 1, \ldots, n &\quad \text{(limit on availability of items).} \end{aligned}$$

The investor goal is thus to determine $x_1, \ldots, x_n$ so as to maximize the profit f_0 while satisfying the above constraints. The described problem is known in the literature as the *knapsack* problem.

Remark 1.1 *A warning on limits of optimization models.* Many, if not all, real-world decision problems and engineering design problems *can*, in principle, be expressed mathematically in the form of an optimization problem. However, we warn the reader that having a problem expressed as an optimization model does not necessarily mean that the problem can then be *solved* in practice. The problem described in Section 1.1.4, for instance, belongs to a category of problems that are "hard" to solve, while the examples described in the previous sections are "tractable," that is, easy to solve numerically. We discuss these issues in more detail in Section 1.2.4. Discerning between hard and tractable problem formulations is one of the key abilities that we strive to teach in this book.

1.2 Optimization problems

1.2.1 Definition

A standard form of optimization. We shall mainly deal with optimization problems[1] that can be written in the following standard form:

$$\begin{aligned} p^* = \min_x \quad & f_0(x) \qquad (1.2) \\ \text{subject to:} \quad & f_i(x) \leq 0, \quad i = 1, \ldots, m, \end{aligned}$$

where

- vector[2] $x \in \mathbb{R}^n$ is the *decision variable*;
- $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the *objective* function,[3] or *cost*;
- $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$, represent the *constraints*;
- p^* is the *optimal value*.

In the above, the term "subject to" is sometimes replaced with the shorthand "s.t.:," or simply by colon notation ":".

Example 1.1 (*An optimization problem in two variables*) Consider the problem

$$\min_x \; 0.9x_1^2 - 0.4x_1x_2 + 0.6x_2^2 - 6.4x_1 - 0.8x_2 : \; -1 \leq x_1 \leq 2, \; 0 \leq x_2 \leq 3.$$

The problem can be put in the standard form (1.2), where:

- the decision variable is $x = (x_1, x_2) \in \mathbb{R}^2$;
- the objective function $f_0 : \mathbb{R}^2 \to \mathbb{R}$, takes values

$$f_0(x) = 0.9x_1^2 - 0.4x_1x_2 - 0.6x_2^2 - 6.4x_1 - 0.8x_2;$$

- the constraint functions $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, 2, 3, 4$ take values

$$f_1(x) = -x_1 - 1, f_2(x) = x_1 - 2, f_3(x) = -x_2, f_4(x) = x_2 - 3.$$

Problems with equality constraints. Sometimes the problem may present explicit *equality* constraints, along with inequality ones, that is

$$\begin{aligned} p^* = \min_x \quad & f_0(x) \\ \text{s.t.:} \quad & f_i(x) \leq 0, \quad i = 1, \ldots, m, \\ & h_i(x) = 0, \quad i = 1, \ldots, p, \end{aligned}$$

where the h_is are given functions. Formally, however, we may reduce the above problem to a standard form with inequality constraints only, by representing each equality constraint via a pair of inequalities. That is, we represent $h_i(x) = 0$ as $h_i(x) \leq 0$ and $h_i(x) \geq 0$.

[1] Often an optimization problem is referred to as a "mathematical program." The term "programming" (or "program") does not refer to a computer code, and is used mainly for historical reasons.

[2] A vector x of dimension n is simply a collection of real numbers $x_1, x_2, \ldots, x_n$. We denote by $\mathbb{R}^n$ the space of all possible vectors of dimension n.

[3] A function f describes an operation that takes a vector $x \in \mathbb{R}^n$ as an input, and assigns a real number, denoted $f(x)$, as a corresponding output value. The notation $f : \mathbb{R}^n \to \mathbb{R}$ allows us to define the input space precisely.

Problems with set constraints. Sometimes, the constraints of the problem are described abstractly via a set-membership condition of the form $x \in \mathcal{X}$, for some subset $\mathcal{X}$ of $\mathbb{R}^n$. The corresponding notation is

$$p^* = \min_{x \in \mathcal{X}} f_0(x),$$

or, equivalently,

$$\begin{aligned} p^* = \min_{x} \quad & f_0(x) \\ \text{s.t.:} \quad & x \in \mathcal{X}. \end{aligned}$$

Problems in maximization form. Some optimization problems come in the form of maximization (instead of minimization) of an objective function, i.e.,

$$p^* = \max_{x \in \mathcal{X}} g_0(x). \tag{1.3}$$

Such problems, however, can be readily recast in standard minimization form by observing that, for any g_0, it holds that

$$\max_{x \in \mathcal{X}} g_0(x) = -\min_{x \in \mathcal{X}} -g_0(x).$$

Therefore, problem (1.3) in maximization form can be reformulated as one in minimization form as

$$-p^* = \min_{x \in \mathcal{X}} f_0(x),$$

where $f_0 = -g_0$.

Feasible set. The *feasible set*[4] of problem (1.2) is defined as

$$\mathcal{X} = \{x \in \mathbb{R}^n \text{ s.t.: } f_i(x) \leq 0, \;\; i = 1, \ldots, m\}.$$

A point x is said to be *feasible* for problem (1.2) if it belongs to the feasible set $\mathcal{X}$, that is, if it satisfies the constraints. The feasible set may be empty, if the constraints cannot be satisfied simultaneously. In this case the problem is said to be *infeasible*. We take the convention that the optimal value is $p^* = +\infty$ for infeasible minimization problems, while $p^* = -\infty$ for infeasible maximization problems.

[4] In the optimization problem of Example 1.1, the feasible set is the "box" in $\mathbb{R}^2$, described by $-1 \leq x_1 \leq 2$, $0 \leq x_2 \leq 3$.

1.2.2 *What is a solution?*

In an optimization problem, we are usually interested in computing the optimal value p^* of the objective function, possibly together with a corresponding *minimizer*, which is a vector that achieves the optimal value, and satisfies the constraints. We say that the problem is attained if there is such a vector.[5]

[5] In the optimization problem of Example 1.1, the optimal value $p^* = -10.2667$ is attained by the optimal solution $x_1^* = 2$, $x_2^* = 1.3333$.

Feasibility problems. Sometimes an objective function is not provided. This means that we are just interested in finding a feasible point, or determining that the problem is infeasible. By convention, we set f_0 to be a constant in that case, to reflect the fact that we are indifferent to the choice of a point x, as long as it is feasible. For problems in the standard form (1.2), solving a feasibility problem is equivalent to finding a point that solves the system of inequalities $f_i(x) \leq 0$, $i = 1, \ldots, m$.

Optimal set. The *optimal set*, or *set of solutions*, of problem (1.2) is defined as the set of feasible points for which the objective function achieves the optimal value:

$$\mathcal{X}_{\text{opt}} = \{x \in \mathbb{R}^n \text{ s.t.: } f_0(x) = p^*, \ \ f_i(x) \leq 0, \ \ i = 1, \ldots, m\}.$$

A standard notation for the optimal set is via the arg min notation:

$$\mathcal{X}_{\text{opt}} = \arg\min_{x \in \mathcal{X}} f_0(x).$$

A point x is said to be *optimal* if it belongs to the optimal set, see Figure 1.2.

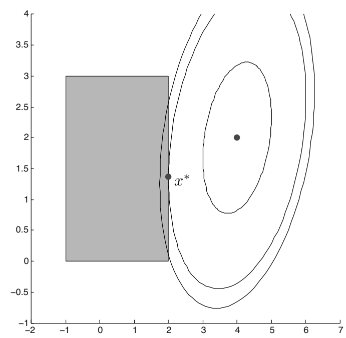

Figure 1.2 A toy optimization problem, with lines showing the points with constant value of the objective function. The optimal set is the singleton $\mathcal{X}_{\text{opt}} = \{x^*\}$.

When is the optimal set empty? Optimal points may not exist, and the optimal set may be empty. This can be for two reasons. One is that the problem is infeasible, i.e., $\mathcal{X}$ itself is empty (there is no point that satisfies the constraints). Another, more subtle, situation arises when $\mathcal{X}$ is nonempty, but the optimal value is only reached in the limit. For example, the problem

$$p^* = \min_x \mathrm{e}^{-x}$$

has no optimal points, since the optimal value $p^* = 0$ is only reached in the limit, for $x \to +\infty$. Another example arises when the constraints include strict inequalities, for example with the problem

$$p^* = \min_x x \ \text{ s.t.: } \ 0 < x \leq 1. \tag{1.4}$$

In this case, $p^* = 0$ but this optimal value is not attained by any x that satisfies the constraints. Rigorously, the notation "inf" should be used instead of "min" (or, "sup" instead of "max") in situations when one doesn't know *a priori* if optimal points are attained. However, in this book we do not dwell too much on such subtleties, and use the min and max notations, unless the more rigorous use of inf and sup is important in the specific context. For similar reasons, we only consider problems with non-strict inequalities. Strict inequalities can be safely replaced by non-strict ones, whenever the objective and constraint functions are continuous. For example, replacing the strict inequality by a non-strict one in (1.4) leads to a problem with the same optimal value $p^* = 0$, which is now attained at a well-defined optimal solution $x^* = 0$.

Sub-optimality. We say that a point x is ϵ-suboptimal for problem (1.2) if it is feasible, and satisfies

$$p^* \leq f_0(x) \leq p^* + \epsilon.$$

In other words, x is ϵ-close to achieving the best value p^*. Usually, numerical algorithms are only able to compute suboptimal solutions, and never reach true optimality.

1.2.3 *Local vs. global optimal points*

A point z is *locally optimal* for problem (1.2) if there exists a value $R > 0$ such that z is optimal for problem

$$\min_x f_0(x) \text{ s.t.: } f_i(x) \leq 0, \;\; i = 1, \ldots, m, \;\; |x_i - z_i| \leq R, \;\; i = 1, \ldots, n.$$

In other words, a local minimizer x minimizes f_0, but only for nearby points on the feasible set. The value of the objective function at that point is *not* necessarily the (global) optimal value of the problem. Locally optimal points might be of no practical interest to the user.

The term *globally optimal* (or optimal, for short) is used to distinguish points in the optimal set $\mathcal{X}_{\text{opt}}$ from local optima. The existence of local optima is a challenge in general optimization, since most algorithms tend to be trapped in local minima, if these exist, thus failing to produce the desired global optimal solution.

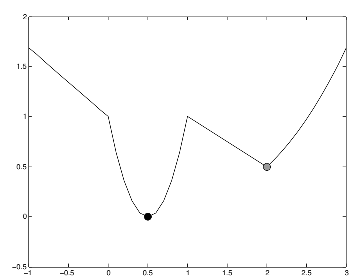

Figure 1.3 Local (gray) vs. global (black) minima. The optimal set is the singleton $\mathcal{X}_{\text{opt}} = \{0.5\}$. The point $x = 2$ is a local minimum.

1.2.4 *Tractable vs. non-tractable problems*

Not all optimization problems are created equal. Some problem classes, such as finding a solution to a finite set of linear equalities or inequalities, can be solved numerically in an efficient and reliable way. On the contrary, for some other classes of problems, no reliable efficient solution algorithm is known.

Without entering a discussion on the *computational complexity* of optimization problems, we shall here refer to as "tractable" all those optimization models for which a globally optimal solution can be found numerically in a reliable way (i.e., always, in any problem instance), with a computational effort that grows gracefully with the *size* of the problem (informally, the size of the problem is measured by the number of decision variables and/or constraints in the model). Other problems are known to be "hard," and yet for other problems the computational complexity is unknown.

The examples presented in the previous sections all belong to problem classes that are tractable, with the exception of the problem in Section 1.1.4. The focus of this book is on tractable models, and a key message is that models that can be formulated in the form of linear algebra problems, or in *convex*[6] form, are typically tractable. Further, if a convex model has some special structure,[7] then solutions can typically be found using existing and very reliable numerical solvers, such as CVX, Yalmip, etc.

[6] See Chapter 8.

[7] See Section 1.3, Chapter 9, and subsequent chapters.

It is also important to remark that tractability is often *not* a property of the problem itself, but a property of our formulation and modeling of the problem. A problem that may seem hard under a certain formulation may well become tractable if we put some more effort and intelligence in the modeling phase. Just to make an example, the raw data in Section 1.1.2 could not be fit by a simple linear model. However, a logarithmic transformation in the data allowed a good fit by a linear model.

One of the goals of this book is to provide the reader with some glimpse into the "art" of manipulating problems so as to model them in a tractable form. Clearly, this is not always possible: some problems are just hard, no matter how much effort we put in trying to manipulate them. One example is the *knapsack* problem, of which the investment problem described in Section 1.1.4 is an instance (actually, most optimization problems in which the variable is constrained to be integer valued are computationally hard). However, even for intrinsically hard problems, for which *exact* solutions may be unaffordable, we may often find useful tractable models that provide us with readily computable *approximate*, or relaxed, solutions.

1.2.5 *Problem transformations*

The optimization formalism in (1.2) is extremely flexible and allows for many transformations, which may help to cast a given problem in a tractable formulation. For example, the optimization problem

$$\min_x \sqrt{(x_1+1)^2 + (x_2-2)^2} \;\; \text{s.t.:} \;\; x_1 \geq 0$$

has the same optimal set as

$$\min_x \left((x_1+1)^2 + (x_2-2)^2\right) \;\; \text{s.t.:} \;\; x_1 \geq 0.$$

The advantage here is that the objective is now differentiable. In other situations, it may be useful to change variables. For example, the problem

$$\max_x \; x_1 x_2^3 x_3 \;\; \text{s.t.:} \;\; x_i \geq 0, \;\; i = 1,2,3, \;\; x_1 x_2 \leq 2, \;\; x_2^2 x_3 \leq 1$$

can be equivalently written, after taking the log of the objective, in terms of the new variables $z_i = \log x_i$, $i = 1,2,3$, as

$$\max_z \; z_1 + 3z_2 + z_3 \;\; \text{s.t.:} \;\; z_1 + z_2 \leq \log 2, \;\; 2z_2 + z_3 \leq 0.$$

The advantage is that now the objective and constraint functions are all linear. Problem transformations are treated in more detail in Section 8.3.4.

1.3 *Important classes of optimization problems*

In this section, we give a brief overview of some standard optimization models, which are then treated in detail in subsequents parts of this book.

1.3.1 *Least squares and linear equations*

A linear least-squares problem is expressed in the form

$$\min_x \sum_{i=1}^{m} \left(\sum_{j=1}^{n} A_{ij} x_j - b_i \right)^2, \qquad (1.5)$$

where A_{ij}, b_i, $1 \leq i \leq m$, $1 \leq j \leq n$, are given numbers, and $x \in \mathbb{R}^n$ is the variable. Least-squares problems arise in many situations, for example in statistical estimation problems such as linear regression.[8]

An important application of least squares arises when solving a set of linear equations. Assume we want to find a vector $x \in \mathbb{R}^n$ such that

$$\sum_{j=1}^{n} A_{ij} x_j = b_i, \;\; i = 1, \ldots, m.$$

Such problems can be cast as least-squares problems of the form (1.5). A solution to the corresponding set of equations is found if the optimal value of (1.5) is zero; otherwise, an optimal solution of (1.5) provides an approximate solution to the system of linear equations. We discuss least-squares problems and linear equations extensively in Chapter 6.

[8] The example in Section 1.1.2 is an illustration of linear regression.