2

The Language of Optimization

2.1 Basic Terms Defined

It is most likely the case that anyone reading a book such as this is familiar with basic definitions of what we are studying, yet it is worthwhile in a mathematical context to offer formal definitions.

Definition 2.1.1 (Maximizer, Maximum)**.** *Let $f : D \to \mathbb{R}$ where the domain D of f is some subset of the real numbers*[1]

- global *(or* absolute*)* maximizer *of $f(x)$ over D if $f(x^*) \geq f(x)$ for all $x \in D$;*
- strict global *(or* strict absolute*)* maximizer *of $f(x)$ over D if $f(x^*) > f(x)$ for all $x \in D$ with $x \neq x^*$;*
- local *(or* relative*)* maximizer *of $f(x)$ if there exists some positive number ϵ such that $f(x^*) \geq f(x)$ for all x, where $x^* - \epsilon < x < x^* + \epsilon$ (i.e. in some* neighborhood *of x^*);*
- strict local *(or* strict relative*)* maximizer *of $f(x)$ if there exists some positive number ϵ such that $f(x^*) > f(x)$ for all x, where $x^* - \epsilon < x < x^* + \epsilon$ with $x \neq x^*$.*

The $f(x^*)$ in the above is, respectively, the global (or absolute) maximum, strict global (or absolute) maximum, local (or relative) maximum, *or* strict local (or relative) maximum *of $f(x)$ over D.*

It is important to understand the difference between a maximizer and a maximum.

Highlight 2.1.2. *The maximum $f(x^*)$ is the optimal value of f which, if the optimal value exists, is unique. The maximizer x^* is the location of the optimal value, which is not necessarily unique.*

A slight change in detail will lead to another important concept:

Definition 2.1.3 (Minimizer, Minimum)**.** *Let $f : D \to \mathbb{R}$ where $D \subseteq \mathbb{R}$. A point x^* in D is said to be a*

[1]It should be noted that we have no need to restrict ourselves to the reals and could offer the definition in a more abstract field.

DOI: 10.1201/9780367425517-2

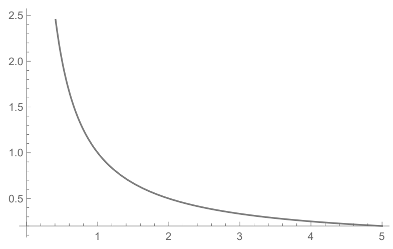

FIGURE 2.1
The graph of $f(x) = \frac{1}{x}$, where $x > 0$.

- global *(or* absolute*)* minimizer *of* $f(x)$ *over* D *if* $f(x^*) \leq f(x)$ *for all* $x \in D$*;*
- strict global *(or* absolute*)* minimizer *of* $f(x)$ *over* D *if* $f(x^*) < f(x)$ *for all* $x \in D$ *with* $x \neq x^*$*;*
- local *(or* relative*)* minimizer *of* $f(x)$ *if there exists some positive number* ϵ *such that* $f(x^*) \leq f(x)$ *for all* x*, where* $x^* - \epsilon \leq x \leq x^* + \epsilon$*;*
- strict local *(or* relative*)* minimizer *of* $f(x)$ *if there exists some positive number* ϵ *such that* $f(x^*) < f(x)$ *for all* x*, where* $x^* - \epsilon < x < x^* + \epsilon$ *with* $x \neq x^*$*.*

The $f(x^*)$ *in the above is, respectively, the* global *(or* absolute*)* minimum*),* strict global minimum*,* local *(or* relative*)* minimum*, or* strict local *(or* relative*)* minimum *of* $f(x)$ *over* D*.*

Note that the plural form of maximum is *maxima* and that the plural form of minimum is *minima.* Together the local and global maxima and minima of a function $f(x)$ are referred to as *extreme values* or *extrema* of $f(x)$. A single maximum or minimum of $f(x)$ is called an *extreme value* or an *extremum* of the function.

We also note that the stated definitions of maximum and minimum are for functions of a single variable, but the definitions[2] are the same for a function of n variables except that $D \subseteq \mathbb{R}^n$ and $\mathbf{x}^* = \langle x_1^*, \ldots, x_n^* \rangle$ would replace x^*.

2.2 When a Max or Min Is Not in the Set

Consider the function $f(x) = \frac{1}{x}$ where $x > 0$. Certainly $f(x)$ is never 0 nor is it ever negative (the graph of $f(x)$ is in Figure 2.1); thus for all $x > 0$, $f(x) > m$ where m is any nonpositive real number. This leads to the following collection of definitions:

[2]There is some concern with how the ϵ interplays with the $x_1, \ldots, x_n$, but the overall idea is the same.

Definition 2.2.1 (Upper and Lower Bounds)**.** *Let f be a function mapping from a set D onto a set R where R is a subset of the real numbers. If there exists a real number M such that $f(x) \leq M$ for all x in D, then f is said to be* bounded from above*. Likewise, if there exists a real number m such that $f(x) \geq m$ for all x in D, then f is said to be* bounded from below*. M is called an* upper bound *of f whereas m is called a* lower bound *of f.*

Example 2.2.2. *The function $f(x) = \frac{1}{x}$ is bounded below by $m = 0$ as well as by $m = -1$. The function is unbounded from above.*

Note that if a function is both bounded above and bounded below, then the function is said to be a *bounded function*; that is

Definition 2.2.3 (Bounded Function)**.** *If there exists a constant M such that $|f(x)| \leq M$ for all x in the domain of f, then f is said to be a bounded function. If no such M exists, then f is said to be* unbounded*.*

Example 2.2.4. *Since $|\sin x| \leq 1$ for any real number x, $\sin x$ is a bounded function.*

Let us now reconsider f in Example 2.2.2 and observe that $f : D \to R$ where $D = (0, \infty) = R$. As previously observed, $f(x) > 0$ for all x in D, but 0 is not in R. As we can find x that get us as arbitrarily close to 0 as we like, f does not have a minimum, but 0 still plays a special role.

Definition 2.2.5 (Infimum, Supremum)**.** *Let S be a nonempty subset of $\mathbb{R}$. Then b is said to be the* infimum *of S if $b \leq s$ for all $s \in S$ and $b \geq m$ for any lower bound m of S. The infimum of a set is the greatest lower bound of the set and is denoted $b = \inf(S)$. Likewise, a is said to be the* supremum *of S if $a \geq s$ for all $s \in S$ and $a \leq M$ for any upper bound M of S. The supremum of a set is the least upper bound of the set and is denoted $a = \sup(S)$.*

It will come as no surprise that the infimum is also often called the *greatest lower bound (glb)*, and the supremum is referred to as the *least upper bound (lub)*. When an infimum or supremum exists, it is unique. As well, the plural of supremum is *suprema* and infimum has *infima* as its plural.

Example 2.2.6. *For f in Example 2.2.2, $\min f$ does not exist, but for the codomain $\mathbb{R}^+$ (the set of positive real numbers), $\inf R = 0$.*

Thus we see that the infimum (when it exists) can play a role similar to a minimum when a minimum does not exist. An analogous statement can be said of maxima and suprema.

2.3 Solving an Optimization Problem

By "solving" algebraic equations like

$$2x^2 + 2x + 5 = 9 \tag{2.1}$$

we mean "finding the particular values of x that satisfy equation 2.1" (they are -2 and 1). In another circumstance, we may be interested in what a lower bound of the polynomial $2x^2 + 2x + 5$ is (this solution is 9/2 or anything smaller).

But when solving an optimization problem, we always mean a little more than just some numeric value. For example, consider the classic algebra problem of a farmer having 1000 feet of fence and wanting to know what is the biggest area he can enclose with a rectangular pen for his livestock if he builds the pen adjacent to his barn (big enough that he only needs fence on three sides). If we label the side parallel to the barn y and the other two sides x, then the mathematical model of our problem is

$$\text{Maximize } \; A = A(x, y) = xy \tag{2.2}$$
$$\text{Subject to } \; y + 2x = 1000 \tag{2.3}$$
$$\text{with } \; x, y \geq 0. \tag{2.4}$$

As our goal is to maximize the area, the function representing it, $A(x, y)$, is called the *objective function*. As well, the amount of available fence puts a restriction on the problem, so the corresponding equation $y + 2x = 1000$ is called a *constraint*. As well, we naturally have the *nonnegativity constraints* that $x, y \geq 0$.

Using the constraint to eliminate a variable, the problem simplifies to

$$\text{Maximize } \; A(x) = 1000x - 2x^2. \tag{2.5}$$

The maximum of this function is 125,000 square feet, and though our farmer will appreciate this knowledge, it is certain he also would like to know what dimensions he needs to make the pen in order to achieve having this maximum area. Thus, by a *solution* to an optimization question, we do not just mean the optimal value of the objective function but also the values of the variables that give the extreme value. Thus we report the answer as $\max A = 125{,}000$ square feet, which occurs when $x = 250$ feet and $y = 500$ feet. We summarize the point of this example in the following Highlight:

Highlight 2.3.1. *A* solution *to an optimization problem is*

1. *the optimal value of the objective function together with*
2. *all possible feasible values of the decision variable(s) that yield the optimal objective value.*

2.4 Algorithms and Heuristics

By an *algorithm* we mean a finite procedure applied to an input with well-defined steps that are repeated to obtain a desired outcome. For example, consider washing your hair. First you wet your hair, then you apply the shampoo and lather, and lastly you rinse. This process may be repeated as many times as you wish to obtain the desired level of cleanliness (read your shampoo bottle; it may have an algorithm written on it). In some sense, an algorithm is a recipe that is repeated.

You may have noticed that we have not offered a formal definition of an algorithm. We are going to avoid unnecessary formality and potential disputes and not offer one all the while noting (modifying Justice Potter Stewart's words in *Jacobellis v. Ohio*:) "I may not know how what the definition of an algorithm is, but I know one when I see it" (Justice Stewart was not addressing algorithms; decency forbids me addressing the matter of that case). It is worthwhile to note that the authoritative text on algorithms – *Algorithms* [11] by Thomas H. Cormen, Charles E. Leiserson, Ronald Rivest, and Clifford Stein – as well does not define the term *algorithm* anywhere in its 1312 pages.

Algorithms have been around for a long time. Perhaps in grade school you learned the *Sieve of Eratosthenes* (circa 3rd century BCE) to find primes. Given a finite list of integers, one circles 2 and crosses out all other multiples of 2. We then proceed to the next available integer, 3, keep it, and cross out all other multiples of 3. We repeat until every integer in our list is circled or crossed out, and what remains are the primes that were in our list of numbers. Algorithms will play a major role in techniques we study iterative methods and combinatorial optimization.

The word *algorithm* has a fascinating origin. It comes from the Latinized ("Algorithmi") version of the Persian name Muḥammad ibn Mūsā al-Khwārizmī whose early 9th century CE book *Al-kitāb al-mukhtaṣar fī ḥisāb al-ğabr wa'l-muqābala* ("The Compendious Book on Calculation by Completion and Balancing") is the first known systematic treatment of algebra as an independent subject. Unlike other early works presenting specific problems and their solution, Al-Khwārizmī's work presents general solution techniques for first- and second-order equations, including completing the square. Al-Khwārizmī can be regarded as the father of algebra, and it is from his text we get the term "algebra" (interestingly, it is also from his name the Spanish and Portuguese get their words for "digit"; see [64]).

Algorithms may produce a globally optimal solution, as we will see in the Simplex Method to solve Linear Programming problems and as well in Kruskal's Algorithm or Prim's Method to find minimum weight spanning trees in a graph. On the other hand, an algorithm may not give a solution but under the right conditions give a good approximation as in Newton's Method.

A *heuristic* is a slightly different monster. A dictionary from before the days of everyday people being familiar with computers would report that "heuristic" is an adjective meaning "enabling a person to discover or learn something for themselves" [15] or "by trial and error" [16]. These days, the word is also regarded as a noun and is most likely shortened from "a heuristic method". When using it as a noun, we mean by *heuristic* a technique that is employed when no method of obtaining a solution (either global or local) is known or a known technique takes too long. It is, in a very true sense, an "educated guess". Consider the *Traveling Salesperson Problem* (TSP) which is introduced in Chapter 25. A salesperson needs to visit a collection of cities and would like to know how to plan her route to minimize distance driven. Unfortunately, there does not yet exist a deterministic-polynomial time algorithm to solve this[3], nor is it known that it is impossible for one to exist ($P = NP$ anyone?), so she can instead do what seems like a good idea: drive to the nearest city and when done with her business there, drive to the nearest city not yet visited, etc. (this is the *Nearest Neighbor Heuristic*[4] that we will see later).

2.5 Runtime of an Algorithm or a Heuristic

A very important matter we will need to consider as we solve our problems is how long it will take a computer to do the work for us. This can be measured in different ways, either time elapsed or the number of operations a computer must perform. Seldom do we use time as the standard in this regard as processor speeds vary and get faster. The standard is to count operations the computer must do, but even this is not precise as sometimes we may count only arithmetic operations performed, but other times we also include calls to memory, etc. This apparent discrepancy is not a large concern as our goal when determining the runtime or *computational complexity*, or simply *complexity*, of an algorithm, heuristic, or computer program is to approximate the amount of effort a computer must put forth. The purpose of these calculations is to compare the runtime efficiency of a given program, algorithm, or heuristic to other known techniques.

Complexity is worthy of its own chapter and is addressed in Chapter 3.

2.6 For Further Study

Parts of these texts have excellent presentations on what we have considered and can be used to deepen one's understanding of the material presented in this chapter:

[3]A brute force algorithm that tests all the paths will give the correct answer and terminate in a finite amount of time. Unfortunately, there are $(n-1)!/2$ possible tours (routes) on n cities, so with 10 cities there are 181,440 possible tours and 20 cities have 60,822,550,204,416,000 possible tours. Hence, though a brute force approach works, we may not live long enough to see the end of the algorithm.

[4]We are referencing specifically the algorithm for "solving" the TSP and not the unrelated algorithm in Machine Learning.

- *An Introduction to Algorithms,* 3rd *edition*, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein; MIT Press (2009). (This is the go-to text for many experts when it comes to the study of algorithms.)
- *Graphs, Algorithms, and Optimization,* 2nd *edition*, William Kocay, Donald L. Kreher, CRC Press (2017)
- *Mathematical Programming An Introduction to Optimization*, Melvyn Jeter, CRC Press (1986)
- *The Mathematics of Nonlinear Programming*, A.L. Peressini, F.E. Sullivan, J.J. Uhl Jr., Springer (1991)

This list is not, of course, an exhaustive list of excellent sources for a general overview of optimization.

2.7 Keywords

(strict) global or absolute maximizer/minimizer, (strict) local or relative maximizer/minimizer, maximum, minimum, infimum, supremum, solution to an optimization problem, algorithm, heuristic, runtime, (computational) complexity.

2.8 Exercises

Exercise 2.1. *State the maximum, minimum, infimum, and supremum (if they exist) of each of the following sets:*

i) $A = \{8, 6, 7, 5, 3, 0, 9\}$,
ii) $B = [a, b)$, *where* $a, b \in \mathbb{R}$,
iii) $C =$ *the range of* $f(x) = 1/(1-x)$, *where* $x \neq 1$,
iv) $D =$ *the range of* $g(x) = 1/(1-x)^2$, *where* $x \neq 1$,
v) $E = \{1 + \frac{(-1)^n}{n}\}$, *where* n *is a positive integer,*
vi) $F =$ *the set of prime numbers.*

Exercise 2.2. *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *and* $\mathbf{x}^* = \langle x_1^*, \ldots, x_n^* \rangle \in \mathbb{R}^n$. *Show* $f(\mathbf{x}^*)$ *is a maximum of* f *if and only if* $-f(\mathbf{x}^*)$ *is a minimum of* $-f$.

Exercise 2.3. *Suppose* s_1 *and* s_2 *are suprema of some set* $S \subset \mathbb{R}$. *Prove* $s_1 = s_2$, *thus establishing that the supremum of a set is unique (obviously, a very similar proof shows that, if it exists, the infimum of a set is also unique).*

Exercise 2.4. *Show if $S \subset \mathbb{R}$ is a nonempty, closed, and bounded set, then* $\sup(S)$ *and* $\inf(S)$ *both belong to S.*

Exercise 2.5. *Let $f : (0, \infty) \to \mathbb{R}$ by $x \mapsto \ln x$ (i.e., $f(x) = \ln x$). Prove that f is monotonically increasing continuous bijection. [Note: this exercise assumes some familiarity with topics in Calculus/Elementary Real Analysis.]*

Exercise 2.6. *Let f be a positive-valued function; that is, its image is a subset of $(0, \infty)$ with $D(f) \subseteq \mathbb{R}$. Prove that $f(x^*) = \max_{x \in D(f)}\{f(x)\}$ if and only if $\ln f(x^*) = \max_{x \in D(f)}\{\ln f(x)\}$ (i.e. the max of f and $\ln f$ occur at the same locations). You may use Exercise 2.5.*