CHAPTER 4: OPTIMIZATION

1 Overview

Optimization is the process of maximizing or minimizing an objective function, subject to constraints. Optimization is fundamental to nearly all academic fields, and its utility extends beyond energy systems and control.

As energy systems engineers, we are often faced with *design* tasks. Here are some examples:

- Determine the optimal battery energy storage capacity for a wind farm [1].
- Optimally place x–MW of photovoltaic generation capacity in Nicaragua to minimize economic cost, yet meet consumer demand.
- Optimally place N vehicle sharing stations in an urban environment to optimally meet user demand [2, 3].
- Optimally re-distribute shared vehicles in a vehicle sharing system to minimize maintenance cost, yet serve user demand [4].
- Design an optimal plug-in electric vehicle charging strategy to minimize economic cost, yet meet mobility demands and limit battery degradation [5].
- Optimally manage energy flow in a smart home, containing photovoltaics, battery storage, and a plug-in electric vehicle, to minimize economic cost [6].
- Optimally manage water flow in Barcelona's water network to meet user demand, maintain sufficient quality, and minimize economic cost [7].

The number of interesting optimization examples in energy systems engineering is limited only by your creativity. These problems often involve physical first principles of an energy storage/conversion device, user demand (a human element), and market economics. In addition, they are almost always too complex to solve with pure intuition. Consequently, one desires a systematic process for designing energy systems that optimizes some metric of interest (e.g. economic cost or performance), while satisfying some necessary constraints (e.g. user demand, physical operating limits, regulations).

1.1 Canonical Form

Nearly all static optimization problems can be abstracted into the following canonical form:

$$\text{minimize} \qquad f(x), \tag{1}$$

$$\text{subject to} \quad g_i(x) \leq 0, \quad i = 1, \cdots, m \tag{2}$$

$$h_j(x) = 0, \quad j = 1, \cdots, l. \tag{3}$$

In this formulation, $x \in \mathbb{R}^n$ is a vector of n decision variables. The function $f(x)$ is known as the "cost function" or "objective function," and maps the decision variables x into a scalar objective function value, mathematically given by $f(x) : \mathbb{R}^n \to \mathbb{R}$. Equation (2) represents m inequality constraints. Each function $g_i(x)$ maps the decision variable to a scalar that must be non-positive for constraint satisfaction, mathematically $g_i(x) : \mathbb{R}^n \to \mathbb{R}, \forall i = 1, \cdots, m$. Similarly, equation (3) represents l equality constraints. Each function $h_i(x)$ maps the decision variable to a scalar that must be zero for constraint satisfaction, mathematically $h_j(x) : \mathbb{R}^n \to \mathbb{R}, \forall j = 1, \cdots, l$. We often vectorize the functions $g_i(x)$ and $h_j(x)$ as $g(x) : \mathbb{R}^n \to \mathbb{R}^m$ and $h(x) : \mathbb{R}^n \to \mathbb{R}^l$, respectively, to more compactly write the canonical form as:

$$\min_x \quad f(x) \tag{4}$$

$$\text{subject to} \quad g(x) \leq 0, \tag{5}$$

$$h(x) = 0. \tag{6}$$

We denote by $x^\star$ the <u>solution</u> that yields the smallest value of $f(x)$ among all the values x that satisfy the constraints. That is, $x^\star$ solves (1)-(3).

Remark 1.1. Note that one can always re-formulate a maximization problem, e.g. $\max_x f(x)$ into a minimization problem by defining $\overline{f}(x) = -f(x)$ and solving $\min_x \overline{f}(x)$.

General optimization problems, without any structure beyond (1)-(3), are often very difficult to solve. Namely, they lack analytical solutions and iterative methods involve tradeoffs. For example, one often sacrifices either long computation times, or not finding the exact solution. However, certain classes of (1)-(3) can be solved efficiently and reliably. Interestingly, huge swaths of practical real-world problems fall within these classes of optimization problems. The crucial skill, then, becomes identifying these types of optimization problems in the real-world.

Several questions or issues arise:

1. What, exactly, is the definition of a minimum?
2. Does a solution even exist, and is it unique?
3. What are the different types of optimization problems?
4. What are the properties of these optimization problems?
5. How do we solve optimization problems?

Throughout this chapter we shall investigate these questions in the context of energy applications.

1.2 Chapter Organization

The remainder of this chapter is organized as follows:

1. Mathematical Preliminaries
2. Linear Programs (LP)
3. Quadratic Programs (QP)
4. Convex Programs (CP)
5. Nonlinear Programs (NLP)

2 Mathematical Preliminaries

Our exposition of optimization begins with some important and useful mathematical concepts. This background will provide the necessary foundation to discuss the theory and algorithms underlying various optimization problems. The first two concepts are *convex sets* and *convex functions*.

2.1 Convex Sets

A set D is convex if the line segment connecting any two points in D completely lies in D. We formalize this concept into the following definition.

Definition 2.1 (Convex Set)**.** *Let D be a subset of $\mathbb{R}^n$. Also, consider scalar parameter $\lambda \in [0,1]$ and two points $a, b \in D$. The set D is convex if*

$$\lambda a + (1-\lambda) b \in D \tag{7}$$

for all points $a, b \in D$.

Figure 1 provides visualizations of convex and non-convex sets. In words, a set is convex if a line segment connecting any two points within domain D is completely within the set D. Figure 2 provides additional examples of convex and non-convex sets.

2.1.1 Examples

The following are some important examples of convex sets you will encounter in optimization:

- The empty set , any single point (i.e. a singleton), $\{x_0\}$, and the whole space $\mathbb{R}^n$ are convex.
- Any line in $\mathbb{R}^n$ is convex.

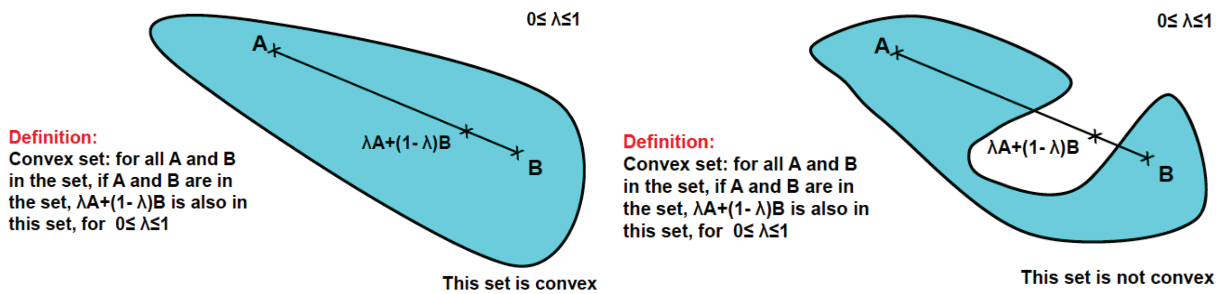


Figure 1: Visualization of convex [left] and non-convex [right] sets.

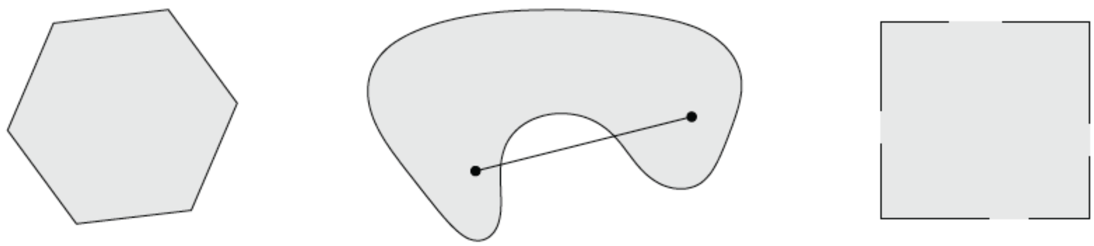


Figure 2: Some simple convex and nonconvex sets. [Left] The hexagon, which includes its boundary (shown darker), is convex. [Middle] The kidney shaped set is not convex, since the line segment between the two points in the set shown as dots is not contained in the set. [Right] The square contains some boundary points but not others, and is not convex.

- Any line segment in $\mathbb{R}^n$ is convex.
- A ray, which has the form $\{x_0 + \theta v \mid \theta \geq 0, v \neq 0\}$ is convex.
- A hyperplane, which has the form $\{x \mid a^T x = b\}$, where $a \in \mathbb{R}^n$, $a \neq 0$, $b \in \mathbb{R}$.
- A halfspace, which has the form $\{x \mid a^T x \leq b\}$, where $a \in \mathbb{R}^n$, $a \neq 0$, $b \in \mathbb{R}$. See Fig. 3.
- A Euclidean ball in $\mathbb{R}^n$, which is centered at x_c and has radius r. Think of the Euclidean ball as a sphere in n-dimensions (See Fig. 4.). Mathematically, the Euclidean ball is

$$B(x_c, r) = \{x \mid \ \|x - x_c\|_2 \leq r\} = \{x \mid (x - x_c)^T (x - x_c) \leq r^2\} \tag{8}$$

- An ellipsoid in $\mathbb{R}^n$, which is centered at x_c. Think of an ellipsoid as an ellipse in n-dimensions. Mathematically,

$$\mathcal{E} = \{x \mid (x - x_c)^T A^{-1} (x - x_c) \leq 1\} \tag{9}$$

where $A = A^T \succ 0$, i.e. A is symmetric and positive definite. The matrix A encodes how the ellipse extends in each direction of $\mathbb{R}^n$. In particular, the length of the semi-axes for the ellipse are given by $\sqrt{\lambda_i(A)}$, where λ_i is the ith eigenvalue of A. See Fig. 5. Another

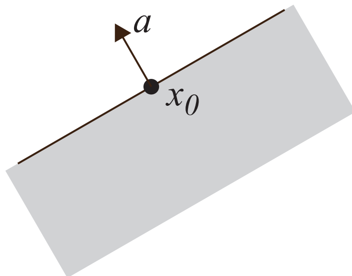


Figure 3: The shaded set is the halfspace given by $\{x \mid a^T(x - x_0) \leq 0\}$ Vector a points in the outward normal direction of the halfspace.

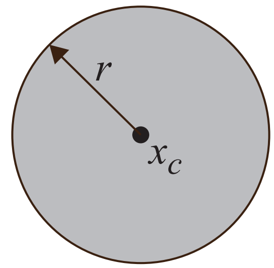


Figure 4: The Euclidean Ball $B(x_c, r)$ is centered at x_c and has radius r. A two-dimensional example is shown.

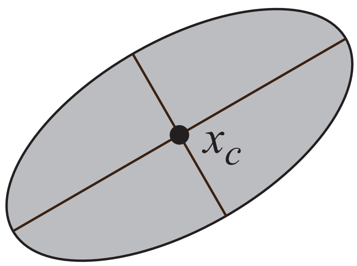


Figure 5: An ellipsoid $\mathcal{E} = \{x_c + Pu \mid \|u\|_2 \leq 1\}$ in two-dimensions with center x_c. The semi-axes have length given by λ_i where $\lambda_i =$eig(P).

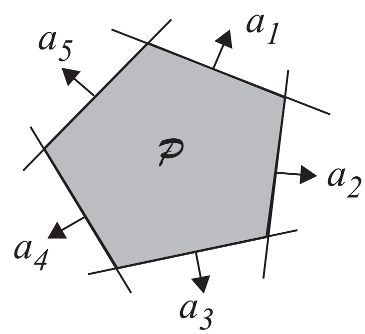


Figure 6: The polyhedron $\mathcal{P}$ is the intersection of five halfspaces, each with outward pointing normal vector a_i.

common representation of an ellipsoid, which we use later in this chapter in Ex 3.4, is:

$$\mathcal{E} = \{x_c + Pu \mid \|u\|_2 \leq 1\} \tag{10}$$

where P is square and positive semi-definite. The semi-axis lengths are given by $\lambda_i(P)$ in this case. If we define $P = A^2$, then this representation is equivalent to (9).

- A polyhedron is defined by the values $x \in \mathbb{R}^n$ that satisfy a finite set of linear inequalities and linear equalities:

$$\mathcal{P} = \left\{x \mid a_i^T x \leq b_i, i = 1, \cdots, m, \;\; a_{eq,j}^T x = b_{eq,j}, j = 1, \cdots, l\right\} \tag{11}$$

See Fig. 6. By definition, a polyhedra is the intersection of a finite number of hyperplanes and halfspaces. Thus, all halfspaces, hyperplanes, lines, rays, and line segments are polyhedra. It is convenient to use the compact vector notation

$$\mathcal{P} = \{x \mid Ax \leq b, \; A_{eq}x = b_{eq}\} \tag{12}$$

Remark 2.1. An interesting property of convex sets is that any convex set can be well-approximated by a polyhedra (198). That is, any convex set $\mathcal{D}$ can be approximated by a finite set of linear inequalities and linear equalities. As the number of linear inequalities and equalities goes to infinity, the approximation error for a general convex set goes to zero.

Exercise 1. *Which of the following sets are convex? Draw each set for the two-dimensional case,* $n = 2$.

(a) A box, i.e., a set of the form $\{x \in \mathbb{R}^n \mid \alpha_i \leq x_i \leq \beta_i, i = 1, \cdots, n\}$.

(b) A slab, i.e., a set of the form $\left\{x \in \mathbb{R}^n \mid \alpha \leq a^T x \leq \beta\right\}$.

(c) A wedge, i.e., $\left\{x \in \mathbb{R}^n \mid a_1^T x \leq b_1, a_2^T x \leq b_2\right\}$.

(d) The union of two convex sets, that is $\mathcal{D}_1 \cup \mathcal{D}_2$*, where* $\mathcal{D}_1, \mathcal{D}_2$ *are convex sets.*

(e) The intersection of two convex sets, that is $\mathcal{D}_1 \cap \mathcal{D}_2$*, where* $\mathcal{D}_1, \mathcal{D}_2$ *are convex sets.*

Exercise 2 (Square and Disk)**.** *Define the square and disk in* $\mathbb{R}^2$ *respectively as*

$$S = \left\{x \in \mathbb{R}^2 \mid 0 \leq x_i \leq 1, i = 1, 2\right\}, \qquad D = \left\{x \in \mathbb{R}^2 \mid \|x\|_2 \leq 1\right\} \tag{13}$$

Are the following statements TRUE or FALSE:

(a) $S \cup D$ *is convex, i.e. the union of sets* S *and* D *is convex*

(b) $S \cap D$ *is convex, i.e. the intersection of sets* S *and* D *is convex*

(c) $S\ D$ *is convex, i.e. set difference of* D *from* S *is convex*

Exercise 3 (Voronoi description of halfspace, [8] p. 60)**.** *Let* a *and* b *be distinct points in* $\mathbb{R}^n$*. Show that the set of all points that are closer (in Euclidean norm) to* a *than* b*, i.e.,* $\{x \mid \|x - a\|_2 \leq \|x - b\|_2\}$*, is a half-space. Describe it explicitly as an inequality of the form* $c^T x \leq d$*. Draw a picture.*

2.2 Convex Functions

Definition 2.2 (Convex Function)**.** *Let* D *be a convex set. Also, consider scalar parameter* $\lambda \in [0, 1]$ *and two points* $a, b \in D$*. Then the function* $f(x)$ *is convex on* D *if*

$$f(x) = f(\lambda a + (1 - \lambda) b) \leq \lambda f(a) + (1 - \lambda) f(b) \tag{14}$$

for all points $a, b \in D$*.*

Definition 2.3 (Concave Function)**.** *Let* D *be a convex set. Also, consider scalar parameter* $\lambda \in [0, 1]$ *and two points* $a, b \in D$*. Then the function* $f(x)$ *is concave on* D *if*

$$f(x) = f(\lambda a + (1 - \lambda) b) \geq \lambda f(a) + (1 - \lambda) f(b) \tag{15}$$

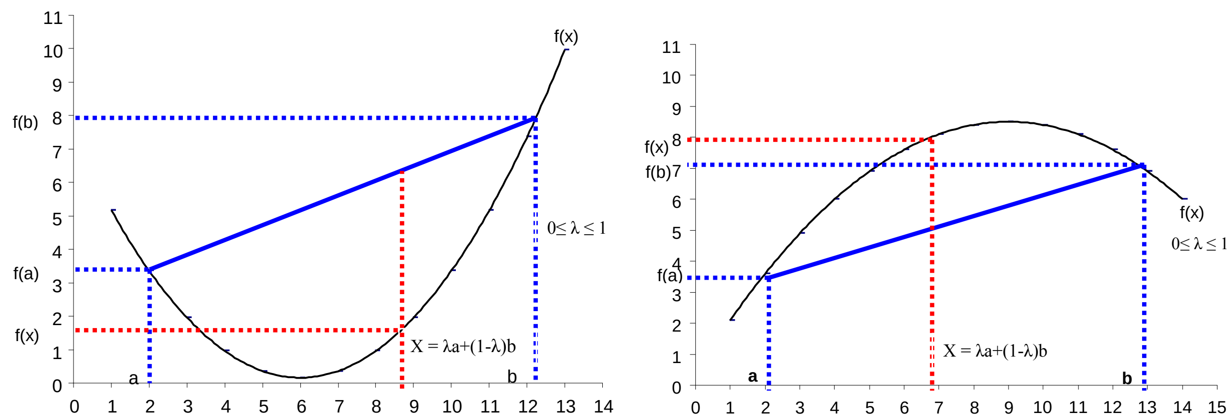


Figure 7: Visualization of convex [left] and concave [right] function definitions.

for all points $a, b \in D$*.*

Figure 7 provides visualizations of the definitions given above. In words, a function is convex if a line segment connecting any two points within domain D is <u>above</u> the function. A function is concave if a line segment connecting any two points within domain D is <u>below</u> the function.

Exercise 4. *Which of the following functions are convex, concave, neither, or both, over the set* $D = [-10, 10]$*? You may use graphical arguments or (14), (15) to prove your claim.*

(a) $f(x) = 0$

(b) $f(x) = x$

(c) $f(x) = x^2$

(d) $f(x) = -x^2$

(e) $f(x) = x^3$

(f) $f(x) = \sin(x)$

(g) $f(x) = e^{-x^2}$

(h) $f(x) = |x|$

Convex and concave functions have several useful properties, summarized by the following proposition.

Proposition 1 (Convex/Concave Function Properties)**.** Consider a function $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ and compact set D.

1. If $f(x)$ is convex on D, then $-f(x)$ is concave on D.
2. If $f(x)$ is concave on D, then $-f(x)$ is convex on D.
3. $f(x)$ is a convex function on $D \iff \frac{d^2 f}{dx^2}(x)$ is positive semi-definite $\forall\ x \ \in D$.
4. $f(x)$ is a concave function on $D \iff \frac{d^2 f}{dx^2}(x)$ is negative semi-definite $\forall\ x \ \in D$.

2.2.1 Examples

It is easy to verify that all linear and affine functions are both convex and concave functions. Here we provide more interesting examples of convex and concave functions. First, we consider functions $f(x)$ where $x \in \mathbb{R}$ is scalar.

- *Quadratic.* $\frac{1}{2}ax^2 + bx + c$ is convex on $\mathbb{R}$, for any $a \geq 0$. It is concave on $\mathbb{R}$ for any $a \leq 0$.
- *Exponential.* e^{ax} is convex on $\mathbb{R}$, for any $a \in \mathbb{R}$.
- *Powers.* x^a is convex on the set of all positive x, when $a \geq 1$ or $a \leq 0$. It is concave for $0 \leq a \leq 1$.
- *Powers of absolute value.* $|x|^p$, for $p \geq 1$ is convex on $\mathbb{R}$.
- *Logarithm.* $\log x$ is concave on the set of all positive x.
- *Negative entropy.* $x \log x$ is convex on the set of all positive x.

Convexity or concavity of these examples can be shown by directly verifying (14), (15), or by checking that the second derivative is non-negative (degenerate of positive semi-definite) or non-positive (degenerate of negative semi-definite). For example, with $f(x) = x \log x$ we have

$$f'(x) = \log x + 1, \qquad f''(x) = 1/x,$$

so that $f''(x) \geq 0$ for $x > 0$. Therefore the negative entropy function is convex for positive x.

We now provide a few commonly used examples in the multivariable case of $f(x)$, where $x \in \mathbb{R}^n$.

- *Norms.* Every norm in $\mathbb{R}^n$ is convex.
- *Max function.* $f(x) = \max\{x_1, \cdots, x_n\}$ is convex on $\mathbb{R}^n$.
- *Quadratic-over-linear function.* The function $f(x, y) = x^2/y$ is convex over all positive x, y.
- *Log-sum-exp.* The function $f(x) = \log\left(\exp^{x_1} + \cdots + \exp^{x_n}\right)$ is convex on $\mathbb{R}^n$. This function can be interpreted as a differentiable (in fact, analytic) approximation of the max function. Consequently, it is extraordinarily useful for gradient-based algorithms, such as the ones described in Section 4.1.
- *Geometric mean.* The geometric mean $f(x) = \left(\Pi_{i=1}^n x_i\right)^{1/n}$ is concave for all elements of x positive, i.e. $\{x \in \mathbb{R}^n \mid x_i > 0 \; \forall \; i = 1, \cdots, n\}$.
- *Log determinant.* The function $f(X) = \log\det(X)$ is concave w.r.t. X for all X positive definite. This property is useful in many applications, including optimal experiment design.

Convexity (or concavity) of these examples can be shown by directly verifying (14), (15), or by checking that the Hessian is positive semi-definite (or negative semi-definite). These are left as exercises for the reader.

2.2.2 Operations that conserve convexity

Next we describe operations on convex functions that preserve convexity. These operations include addition, scaling, and point-wise maximum. Often, objective functions in the optimal design of engineering system are a combination of convex functions via these operations. This section helps you analyze when the combination is convex, and how to construct new convex functions.

Linear Combinations

It is easy to verify from (14) that when $f(x)$ is a convex function, and $\alpha \geq 0$, then the function $\alpha f(x)$ is convex. Similarly, if $f_1(x)$ and $f_2(x)$ are convex functions, then their sum $f_1(x) + f_2(x)$ is a convex function. Combining non-negative scaling and addition yields a non-negative weighted sum of convex functions

$$f(x) = \alpha_1 f_1(x) + \cdots + \alpha_m f_m(x) \tag{16}$$

that is also convex.

Pointwise Maximum

If $f_1(x)$ and $f_2(x)$ are convex functions on $\mathcal{D}$, then their *point-wise maximum* f defined by

$$f(x) = \max\{f_1(x), f_2(x)\} \tag{17}$$

is convex on $\mathcal{D}$. This property can be verified via (14) by considering $0 \leq \lambda \leq 1$ and $a, b \in \mathcal{D}$.

$$\begin{aligned}
f(\lambda a + (1-\lambda)b) &= \max\left\{f_1(\lambda a + (1-\lambda)b), f_2(\lambda a + (1-\lambda)b)\right\} \\
&\leq \max\left\{\lambda f_1(a) + (1-\lambda) f_1(b), \lambda f_2(a) + (1-\lambda) f_2(b)\right\} \\
&\leq \lambda \max\left\{f_1(a), f_2(a)\right\} + (1-\lambda) \max\left\{f_1(b), f_2(b)\right\} \\
&= \lambda f(a) + (1-\lambda) f(b).
\end{aligned}$$

which establishes convexity of f. It is straight-forward to extend this result to show that if $f_1(x), \cdots, f_m(x)$ are convex, then their point-wise maximum

$$f(x) = \max\{f_1(x), \cdots, f_m(x)\} \tag{18}$$

is also convex.

Composition with an Affine Mapping

Consider $f(\cdot) : \mathbb{R}^n \to \mathbb{R}$ with parameters $A \in \mathbb{R}^{n\times m}$ and $b \in \mathbb{R}^n$. Define function $g(\cdot) : \mathbb{R}^m \to \mathbb{R}$ as

$$g(x) = f(Ax + b) \tag{19}$$

If f is convex, then g is convex. If f is concave, then g is concave.

Exercise 5 (Simple function compositions)**.** *Consider function $g(\cdot) : \mathbb{R} \to \mathbb{R}$ defined over convex set $\mathcal{D}$. Prove each of the following function compositions is convex or concave over $\mathcal{D}$.*

(a) If g is convex, then $\exp g(x)$ is convex.

(b) If g is concave and positive, then $\log g(x)$ is concave.

(c) If g is concave and positive, then $1/g(x)$ is convex.

(d) If g is convex and nonnegative and $p \geq 1$, then $g(x)^p$ is convex.

(e) If g is convex then $-\log(-g(x))$ is convex on $\{x|g(x) < 0\}$.

2.3 Definition of Minimizers

Armed with notions of convex sets and convex/concave functions, we are positioned to provide a precise definition of a minimizer, which we often denote with the "star" notation as x^*. There exist two types of minimizers: global and local minimizers. Their definitions are given as follows.

Definition 2.4 (Global Minimizer)**.** *$x^* \in D$ is a global minimizer of $f(x)$ on D if*

$$f(x^*) \leq f(x), \qquad \forall\, x \in D \tag{20}$$

In words, this means x^* minimizes $f(x)$ everywhere in D. In contrast, we have a local minimizer.

Definition 2.5 (Local Minimizer)**.** *$x^* \in D$ is a local minimizer of $f(x)$ on D if*

$$\exists\, \epsilon > 0 \quad \textit{s.t.} \quad f(x^*) \leq f(x), \qquad \forall\, x \in D \cap \{x \in \mathbb{R} \mid \|x - x^*\| < \epsilon\} \tag{21}$$

In words, this means x^* minimizes $f(x)$ locally in D. That is, there exists some neighborhood whose size is characterized by ϵ where x^* minimizes $f(x)$. Examples of global and local minimizers are provided in Fig. 8

We now have a precise definition for a minimum. However, we now seek to understand when a minimum even exists. The answer to this question leverages the convex set notion, and is called the Weierstrauss Extreme Value Theorem.

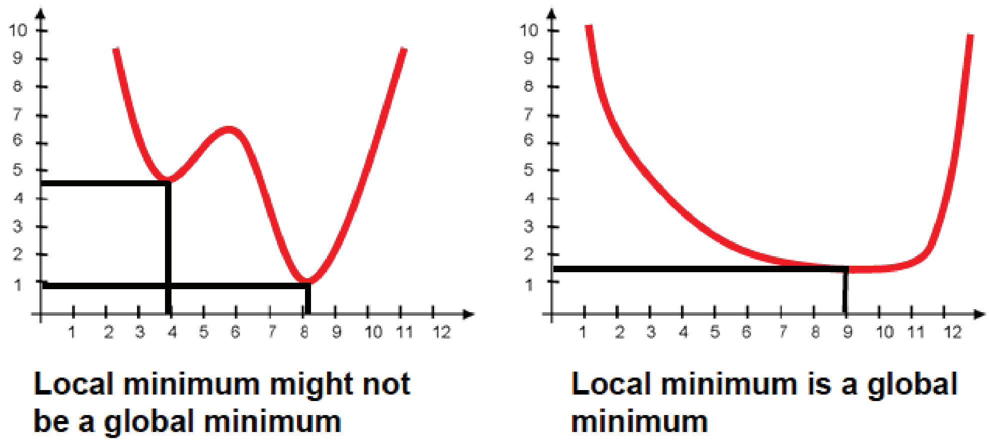


Figure 8: The LEFT figure contains two local minimizers, but only one global minimizer. The RIGHT figure contains a local minimizer, which is also the global minimizer.

Theorem 2.1 (Weierstrass Extreme Value Theorem)**.** *If $f(x)$ is continuous and bounded on a convex set D, then there exists at least one global minimum of f on D.*

A visualization of this theorem is provided in Fig. 9. In practice, the result of the Weierstrauss extreme value theorem seems obvious. However, it emphasizes the importance of having a continuous and bounded objective function $f(x)$ from (1), and constraints (2)-(3) that form a convex set. Consequently, we know a global minimizer exists if we strategically formulate optimization problems where the objective function is continuous and bounded, and the constraint set is convex.

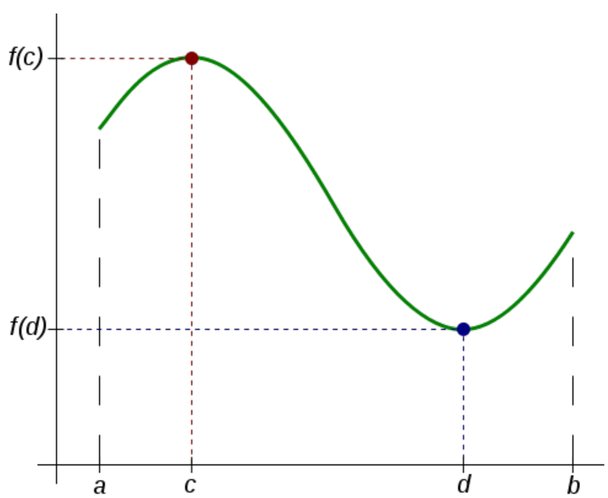


Figure 9: In this graph, $f(x)$ is continuous and bounded. The convex set is $D = [a, b]$. The function f attains a global minimum at $x = d$ and a global maximum at $x = c$.

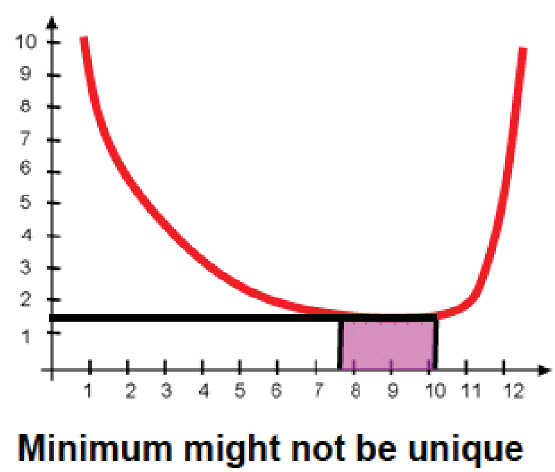


Figure 10: A local or global minimum need not be unique.

Is the minimum unique? In general, the minimum need not be unique, as illustrated in Fig. 10. There may be two optima or even infinite optima. The physical interpretation is that a multitude of designs produce equally good solutions, in terms of the objective function value.

2.4 A Zoology of Optimization Programs

Armed with a background in convex sets and convex functions, we are positioned to classify different types of optimization programs. In fact, there exists an entire zoology of optimization programs. In this chapter, we discuss the following types:

- Linear program (LP)
- Quadratic program (QP)
- Convex program (CP)
- Nonlinear program (NLP)

These classes of optimization problems are not mutually exclusive – some sets subsume others. For example, all LPs are CPs. Another example is that all LPs, QPs, and CPs, are NLPs. See Fig. 11. The remainder of this chapter is dedicated to understanding properties of these various optimization program types.

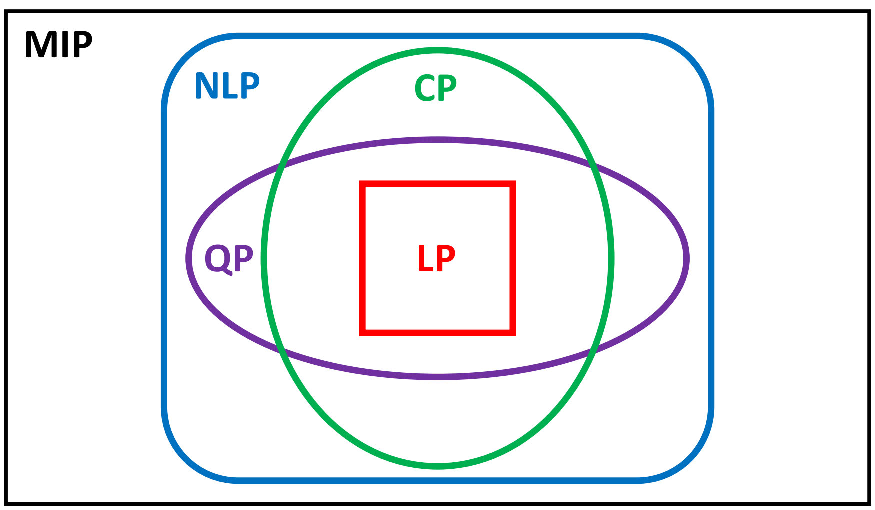


Figure 11: Optimization programs fall into various classes: linear programs (LP), quadratic programs (QP), convex programs (CP), nonlinear programs (NLP), and mixed integer programs (MIP). In these course notes, we focus on LP, QP, CP, and NLP.

3 Convex Programming

3.1 Definition

A *convex optimization problem* has the form

$$\begin{aligned} \text{minimize} \quad & f(x) && (22)\\ \text{subject to} \quad & g_i(x) \leq 0, \quad i = 1, \cdots, m && (23)\\ & a_j^T x = b_j, \quad j = 1, \cdots, l. && (24) \end{aligned}$$

Comparing this problem with the abstract optimization problem in (1)-(3), the *convex optimization problem* has three additional requirements:

- objective function $f(x)$ must be convex,
- the inequality constraint functions $g_i(x)$ must be convex for all $i = 1, \cdots, m$,
- the equality constraint functions $h_j(x)$ must be affine for all $j = 1, \cdots, l$.

Note that in the convex optimization problem, we can only tolerate affine equality constraints, meaning (24) takes the matrix-vector form of $A_{eq}x = b_{eq}$.

In general, no analytical formula exists for the solution of convex optimization problems. However, there are very effective and reliable methods for solving them. For example, we can easily solve problems with hundreds of variables and thousands of constraints on a current laptop computer, in at most a few tens of seconds. Due to the impressive efficiency of these solvers, many researchers have developed tricks for transforming problems into convex form. As a result, a surprising number of practical energy system problems can be solved via convex optimization. With only a bit of exaggeration, we can say that, if you formulate a practical problem as a convex optimization problem, then you have solved the original problem. Recognizing a convex optimization problem can be difficult, however. The challenge, and art, in using convex optimization is in recognizing and formulating the problem. Once this formulation is done, solving the problem is essentially an off-the-shelf technology.

Example 3.1. Are the following two problems convex programs?

$$\begin{aligned} \text{minimize} \quad & f(x) = x_1^2 + x_2^2 && (25)\\ \text{subject to} \quad & g_1(x) = x_1/(1 + x_2^2) \leq 0 && (26)\\ & h_1(x) = (x_1 + x_2)^2 = 0 && (27) \end{aligned}$$

This is NOT a convex program. The inequality constraint function $g_1(x)$ is not convex in (x_1, x_2). Additionally, the equality constraint function $h_1(x)$ is not affine in (x_1, x_2).

Now, an astute observer might comment that both sides of (26) can be multiplied by $(1 + x_2^2)$ and (27) can be represented simply by $x_1 + x_2 = 0$, without loss of generality. This leads to:

$$\text{minimize} \quad f(x) = x_1^2 + x_2^2 \tag{28}$$
$$\text{subject to} \quad g_1(x) = x_1 \leq 0 \tag{29}$$
$$h_1(x) = x_1 + x_2 = 0 \tag{30}$$

This is a convex program. The objective function $f(x)$ and inequality constraint function $g_1(x)$ are convex in (x_1, x_2). Additionally, the equality constraint function $h_1(x)$ is affine in (x_1, x_2).

The remainder of this chapter is dedicated to examples. First, let us discuss some properties of convex programs.

3.2 Properties

The following statements are true about convex programming problems:

- If a local minimum exists, then it is the global minimum.
- If the objective function is strictly convex, and a local minimum exists, then it is a unique minimum.

The implication of these properties is stunning. The first property states that you need only find a local minimum (if it exists). In other words, if you can prove that a candidate solution x is optimal over a local neighborhood around x, then you are done. There is no need to search elsewhere. That local optimizer $x^\star$ is globally optimal.

The second property is also important. If in addition to finding a local minimum, you observe that the objective function is strictly convex (that is, it satisfies (14) with a strict inequality), then that local minimum is unique. That means there exists no other local minimizer $x^\star$ that does equally well. You have found the one and only optimal solution.

Next we present a series of important types of convex programs. These examples persistently arise in energy systems optimization, along with applications ranging from transportation, environmental science & engineering, computer science, structural design, etc. The important examples include:

- Linear programming (LP)
- Quadratic programming (QP)
- Geometric programming (GP)
- Second order cone programming (SOCP)
- Maximum likelihood estimation (MLE)

3.3 Linear Programming

A linear program (LP) is defined as the following special case of a convex program:

$$\begin{aligned} \text{minimize} \quad & c^T x && (31) \\ \text{subject to} \quad & Ax \leq b && (32) \\ & A_{eq}x = b_{eq} && (33) \end{aligned}$$

Compare with the canonical form (1)-(3). We find that $f(x)$ must be linear (or affine, before dropping the additive constant). Also, $g_i(x)$ and $h_j(x)$ must be affine for all i and j, respectively.

Appendix Section 5 provides a more detailed exposition of LPs. However, we highlight the most important properties here. Observe that the constraint set $\{x \in \mathbb{R}^n \mid Ax \leq b, A_{eq}x = b_{eq}\}$ forms a polyhedron – one of the important convex set examples discussed in Section 2.1. An example for $n = 2$ is visualized in Fig. 12. In addition, observe that the objective function is linear, meaning the sets of equivalent cost form "isolines" across the Cartesian plane. In Fig. 12, the cost decreases as we move from left to right.

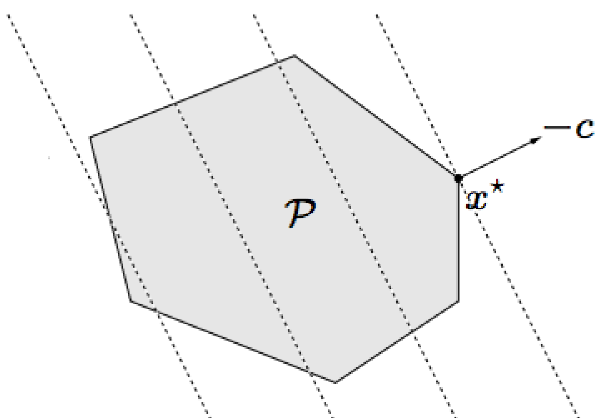


Figure 12: The feasible set of LPs always forms a polyhedron $\mathcal{P}$. The objective function can be visualized as isolines with constant cost, visualized by the dotted lines. The optimal solution is at the boundary point that touches the isoline of least cost.

Consequently, the visual approach to solving LPs is the following: Find the point in the feasible set (which is a polyhedron $\mathcal{P}$) that touches the isoline with least value. Visually, it is obvious the optimum must be bounded (if it is feasible). In other words, the minimizer $x^\star$ always exists on the boundary of the polyhedron (if it is feasible). This observation leads to the following proposition, provided without proof.

Proposition 2 (LP Solutions)**.** The solution to any linear program is characterized by one of the following three categories:

- **[No Solution]** This occurs when the feasible set is empty, or the objective function is unbounded.
- **[One Unique Solution]** There exists a single unique solution at the vertex of the feasible set. That is, at least two constraints are active and their intersection gives the optimal solution.
- **[A Non-Unique Solution]** There exists an infinite number of solutions, given by one edge of the feasible set. That is, one or more constraints are active and all solutions along the intersection of these constraints are equally optimal. This can only occur when the objective function gradient is orthogonal to one or multiple constraint.

Given this foundation, we now dive into examples of LPs in energy systems optimization.

Example 3.2 (Optimal Economic Dispatch)**.** Imagine you are the California Independent System Operator (CAISO). Your role is to schedule power plant generation for tomorrow. Specifically, there are n generators, and you must determine how much power each generator produces during each one hour increment, across 24 hours. The power generated must equal the power consumed. Moreover, you seek to contract these generators in the most economic fashion possible. The following data is given:

- Each generator indexed i provides its "marginal cost" c_i (units of USD/MW). Quantity c_i indicates the financial compensation each generator requests for providing one unit of power.
- Each generator has a maximum power capacity of $x_{i,\max}$ (units of MW). You may not contract more power than each generator can produce.
- The electricity demand for California is $D(k)$, where k indexes each hour, i.e. $k = 0, 1, \cdots, 23$.

We are now positioned to formulate a LP that can be solved with convex solvers:

$$\begin{aligned}
\text{minimize} \quad & \sum_{k=0}^{23} \sum_{i=1}^{n} c_i x_i(k) && (34)\\
\text{subject to} \quad & 0 \leq x_i(k) \leq x_{i,\max}, \quad \forall\, i = 1, \cdots, n, \quad k = 0, \cdots, 23 && (35)\\
& \sum_{i=1}^{n} x_i(k) = D(k), \quad k = 0, \cdots, 23 && (36)
\end{aligned}$$

where the optimization variable is $x_i(k)$ and physically represents the power produced by generator i during hour k. Example economic dispatch data and results are provided in Fig. 13 and 14.

Several notable properties enable an intuitive view of the solution. First, the optimization problems are independent across time. Therefore, optimal economic dispatch can be solved independently for each time k. That is, solve for each time step k:

$$\text{minimize} \quad \sum_{i=1}^{n} c_i x_i(k) \qquad (37)$$

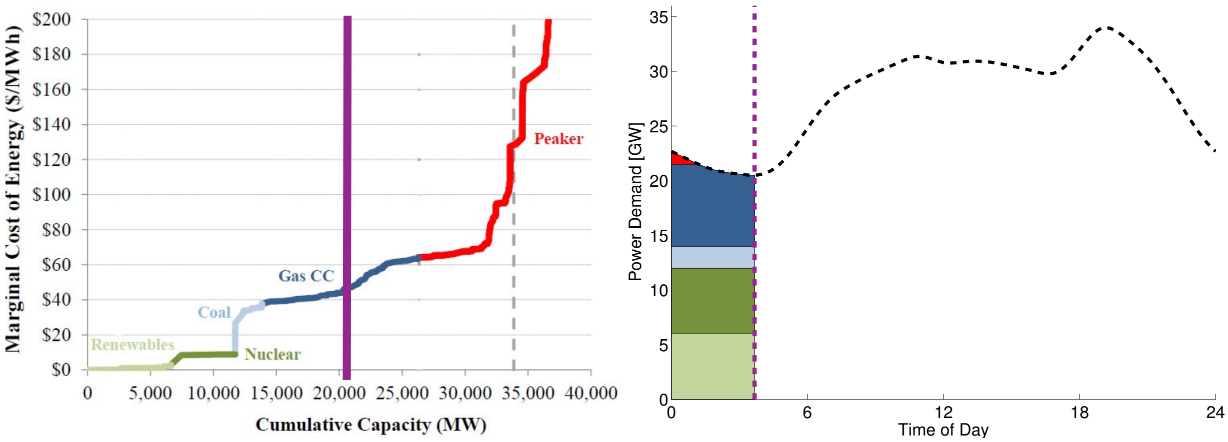


Figure 13: [LEFT] Marginal cost of electricity for various generators, as a function of cumulative capacity. The purple line indicates the total demand $D(k)$. All generators left of the purple line are dispatched. [RIGHT] Optimal supply mix and demand for 03:00.

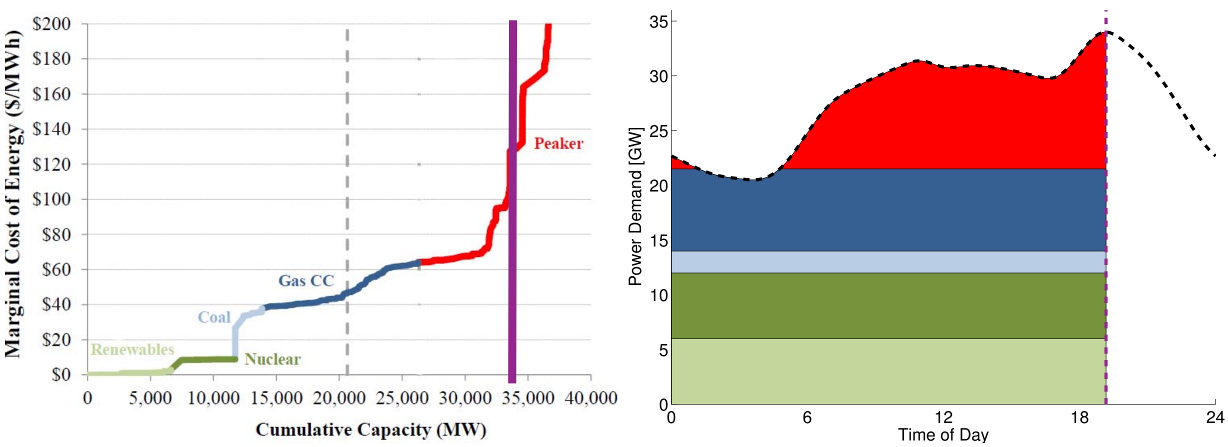


Figure 14: [LEFT] Marginal cost of electricity for various generators, as a function of cumulative capacity. The purple line indicates the total demand $D(k)$. All generators left of the purple line are dispatched. [RIGHT] Optimal supply mix and demand for 19:00.

$$\text{subject to} \quad 0 \leq x_i(k) \leq x_{i,\max}, \quad \forall\, i = 1, \cdots, n, \quad k = 0, \cdots, 23 \tag{38}$$

$$\sum_{i=1}^{n} x_i(k) = D(k), \quad k = 0, \cdots, 23 \tag{39}$$

After solving 24 independent problems, one obtains the minimum generation cost for each hour denoted $f(x^\star(k)) = \sum_{i=1}^n c_i x_i^\star(k)$. Then the minimum daily generation cost is the sum of minimal hourly generation costs: $\sum_{k=0}^{23} f(x^\star(k))$.

Second, the LP has a specific structure that is often called a “knapsack problem” or a “water-filling problem”. In this case, we sort the marginal costs in increasing order and compute the cumulative power capacity of the generators. This process yields the [LEFT] plots in Fig. 13 and 14. Then we observe the electricity demand, such as 20,000 MW at 03:00 in Fig. 13. In the cumulative capacity vs marginal cost plot, the optimal solution is to dispatch the generators to the left of the electricity demand. For example, at 03:00 the maximum capacity of renewables, nuclear, and coal is dispatched, and the remaining electricity is delivered by gas combined cycle (CC) plants.

This, of course, is a stylized and simplified version of real-world optimal economic dispatch in power systems. In real power systems, electricity is delivered over a network with various network constraints. For example, transmission lines and transformers have power limits to ensure safety. Real-world optimal economic dispatch includes network constraints, that incorporate up to thousands of nodes. This renders a considerably more difficult optimization problem that is almost never a LP. However, much recent research has focused on clever reformulations into convex programs.

3.4 Quadratic Programming

A quadratic program (QP) is defined as the following:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}x^T Q x + R^T x + S && (40) \\ \text{subject to} \quad & Ax \leq b && (41) \\ & A_{eq} x = b_{eq} && (42) \end{aligned}$$

Compare with the canonical form (1)-(3). We find that $f(x)$ must be quadratic. However, $g_i(x)$ and $h_j(x)$ are still restricted to be affine for all i and j, respectively.

The QP described above is not always a convex program. Please see Fig. 11. Some classes of QPs are not convex programs. If Q is positive semi-definite, i.e. $Q \succeq 0$, then the QP above is a "convex quadratic program". If Q is not positive semi-definite, then we call this a "nonconvex QP". In this section, we restrict our attention to convex QPs. Appendix Section 6 provides a more detailed exposition of QPs.

A generalization of QPs is quadratically constrained quadratic programs (QCQPs), given by:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}x^T Q x + R^T x + S && (43) \\ \text{subject to} \quad & \frac{1}{2}x^T Q_i x + R_i^T x + S_i \leq 0, \quad \forall\, i = 1, \cdots, m && (44) \\ & A_{eq} x = b_{eq} && (45) \end{aligned}$$

where $Q, Q_i \succeq 0$ for the program to be convex. Compare with the canonical form (1)-(3). We find that $f(x)$ must be quadratic and now the $g_i(x)$ inequality constraint functions may be quadratic for all i. The functions $h_j(x)$ must be affine for all j.

Example 3.3 (Markowitz Portfolio Optimization)**.** Imagine you are an investment portfolio manager. You control a large sum of money, which can be invested in up to n assets or stocks. At the end of some time period, your investment produces a financial return. The key challenge, here, is this return is not predictable. It is random.

Denote by x_i the fraction of funds invested into asset i. Consequently, $\sum_i x_i = 1$ and we assume $x_i \geq 0$, meaning we cannot invest a negative amount into asset i. (Negative x_i is called a "short position", and implies we are obligated to buy asset i at the end of the investment period.)

Assume the return on each asset can be well-characterized by a multivariate Gaussian distribution. Specifically, the return is distributed according to $\mathcal{N}(\mu, \Sigma)$, where $\mu \in \mathbb{R}^n$ is the expected return and $\Sigma \in \mathbb{R}^{n \times n}$ is the covariance. For example, asset i may have an expected return of $\mu_i =$ 2% with a standard deviation of $\sqrt{\Sigma_{ii}} =$ 5%. In contrast, asset j might have an expected return of $\mu_j =$ 5%, but a standard deviation of $\sqrt{\Sigma_{jj}} =$ 50%. Would it be wise to invest everything into asset j? Would it be wise to invest everything into asset i? What about some mix?

This is the classical portfolio optimization problem, introduced by Markowitz. It is mathematically given as:

$$\text{minimize} \quad x^T \Sigma x \tag{46}$$

$$\text{subject to} \quad \mu^T x \geq r_{\min} \tag{47}$$

$$\mathbb{1}^T x = 1, \quad x \succeq 0 \tag{48}$$

In words, we seek to minimize the return variance while guaranteeing a minimum expected return of $r_{\min}$. The final constraint is a budget constraint. This problem is clearly a QP, and can be solved via convex solvers. Observe that we are minimizing risk, subject to achieving sufficiently high expected return. What if we reverse these roles?

Let us consider maximizing expected return, subject to an upper bound on allowed risk. We write:

$$\text{minimize} \quad -\mu^T x \tag{49}$$

$$\text{subject to} \quad x^T \Sigma x \leq R_{\max} \tag{50}$$

$$\mathbb{1}^T x = 1, \quad x \succeq 0 \tag{51}$$

where $R_{\max}$ represents the maximum allowable risk. Observe that this problem is a QCQP, which follows the form in (43)-(45). This problem can also be solved by convex solvers.

One common variation is to form a bi-criterion problem. Namely, maximize the expected return and minimize the return variance. In math:

$$\text{minimize} \quad x^T \Sigma x \quad \text{AND} \quad \text{maximize} \quad \mu^T x \tag{52}$$

Of course, these two objectives cannot be achieved without tradeoffs. Therefore, one often "scalarizes" this bi-criterion problem to explore the tradeoff:

$$\text{minimize} \quad -\mu^T x + \gamma \cdot x^T \Sigma x \tag{53}$$
$$\text{subject to} \quad \mathbb{1}^T x = 1, \quad x \succeq 0 \tag{54}$$

where the parameter $\gamma \geq 0$ is called the "risk aversion" parameter. As you increase γ, you become more sensitive to variance in your investment returns. A value of $\gamma = 0$ classifies you as a pure risk seeker. You ignore variances and simply pursue the highest expected return.

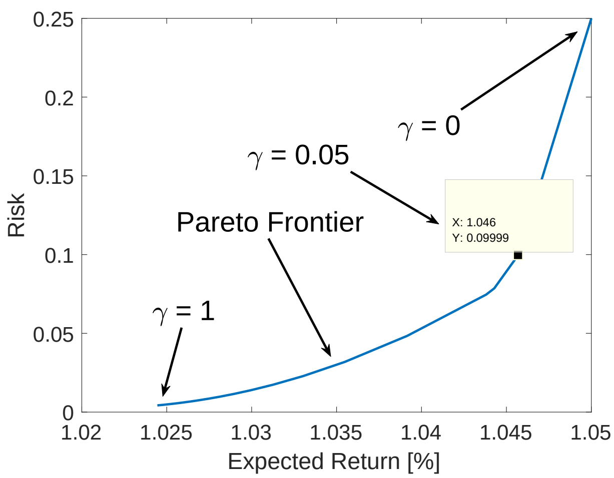


Figure 15: Trade off between maximizing expected return and minimizing risk. This trade off curve is called a "Pareto Frontier"

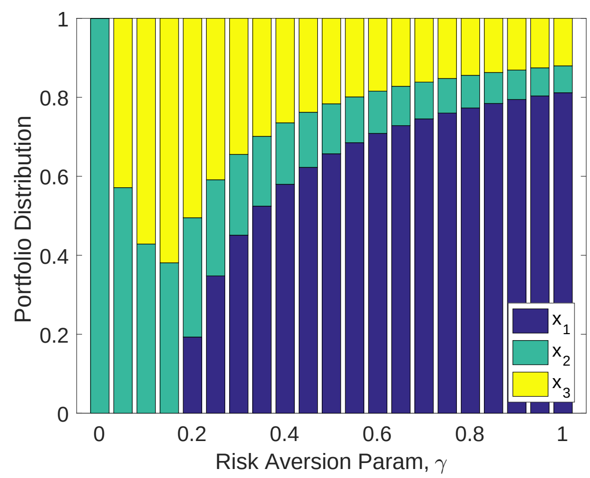


Figure 16: Optimal portfolio investment strategy, as risk aversion parameter γ increases.

To illustrate, consider the following expected return and covariance data for a portfolio of three assets:

$$\mu = [1.02, \; 1.05, \; 1.04]^T, \quad \Sigma = \begin{bmatrix} (0.05)^2 & 0 & 0 \\ 0 & (0.5)^2 & 0 \\ 0 & 0 & (0.1)^2 \end{bmatrix} \tag{55}$$

Asset 1 has a 2% expected return and 5% standard deviation. In contrast, asset 2 has a larger expected return (5%) but significantly larger volatility: 50% standard deviation. What is the optimal investment strategy that balances expected return and risk?

Let us sweep risk aversion parameter between $\gamma = 0$ and $\gamma = 1$. (Note: one can increase γ beyond one). Figure 15 demonstrates the trade off between maximizing expected return versus minimizing risk. Clearly, we seek high expected return and low risk. Therefore, solutions toward the bottom-right are desirable. However, there is a fundamental trade off. When $\gamma = 0$, we ignore risk. The expected return is relatively high, but also imputes high risk. Increasing γ reduces the imputed risk, but also decreases the expected return. For example, when $\gamma = 1$ then we've reduced the risk by 10X, but also reduced the expected return from 5% to 2.5%.

Let us highlight $\gamma = 0.05$ as an interesting solution. For $\gamma = 0.05$, we have an expected return of 4.6% and risk of 0.1. Relative to the $\gamma = 0$ solution, we reduced risk by 60%, but only sacrificed a meager 0.4% in expected return. This trade off could make excellent sense. Namely, it is absolutely worthwhile to reduce volatility by 60% if I only sacrifice 0.4% in expected returns. This example highlights how to use Pareto frontiers. We look for "knees" or "kinks" in the Pareto frontier where it's possible gain a lot in one objective while giving up little in the other.

Consider Fig. 16, which visualizes the optimal portfolio investment strategy as a function of increasing γ. When $\gamma = 0$, we ignore risk. Consequently, the optimal solution is to invest everything in asset 2. It has highest expected return (5%). As we increase γ, we become more sensitive to risk. Consequently, the optimal portfolio diversifies. Initially, we invest in both asset 2 and 3. Asset 3 has the next highest expected return. For sufficiently high γ, we begin to strongly favor asset 1 because it has the lowest variance. As we increase $\gamma \to \infty$, then the optimal portfolio investment strategy would be to invest in asset 1 only, due to low variance.

3.5 Second Order Cone Programming

A second order cone program (SOCP) is defined as the following special case of a convex program:

$$\begin{aligned} \text{minimize} \quad & f^T x && (56)\\ \text{subject to} \quad & \|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad i = 1, \cdots, m && (57)\\ & A_{eq} x = b_{eq} && (58) \end{aligned}$$

The inequalities are called second order cone constraints. Indeed, they form a special class of convex sets called "second order cones". While this second order cone constraint may seem highly technical, specific, and abstract, it can be the crucial feature that enables us to study – for example – optimal dispatch of renewable generators under uncertainty.

Example 3.4 (Robust Dispatch in a High-Penetration Renewable Grid)**.** Let us return to the economic dispatch problem. We must economically dispatch up to n generators to serve electricity demand D. In this case, we consider that a large percentage of these n generators are renewable. The challenge we face is the following. The maximum power generating capacity of these renewable generators is random. For example, a wind farm can produce anywhere between 0 MW and 10 MW, depending on weather conditions. Similarly, a solar farm can produce anywhere from 0 MW to 20 MW.

To model this scenario, we write the following linear program

$$\begin{aligned} \text{minimize} \quad & f^T x && (59)\\ \text{subject to} \quad & R^T x \geq D && (60) \end{aligned}$$

$$0 \leq x \leq 1 \tag{61}$$

where $x \in \mathbb{R}^n$ is the vector of power generation dispatched to the generators, as a fraction of the generator's rated capacity. For example, to dispatch 9 MW from the wind farm rated at 10 MW, we set $x_i = 0.9$. To dispatch 15 MW from the solar farm rated at 20 MW, we set $x_j = 0.75$. Parameter $f \in \mathbb{R}^n$ is the vector of marginal costs, and $D \in \mathbb{R}$ is the electricity demand. Parameter $R \in \mathbb{R}^n$ represents the real-time power capacity of the generators. To convert (59)-(60) into standard form, we define $a = -R$ and $b = -D$, which yields

$$\begin{aligned} \text{minimize} \quad & f^T x && (62)\\ \text{subject to} \quad & a^T x \leq b && (63)\\ & 0 \leq x \leq 1 && (64) \end{aligned}$$

Our primary focus is on parameter a. While parameters f and b are fixed, parameter a is uncertain. In particular, if generator i is the wind farm, then R_i might vary between 0 MW and 10 MW (thus $-10 \leq a_i \leq 0$). If generator j is the solar farm, then R_j might vary between 0 MW and 20 MW (thus $-20 \leq a_j \leq 0$). Finally, if generator k is a natural gas plant with 50 MW capacity, then R_k is always 50 MW (thus $a_k = -50$). Mathematically, we hypothesize that vector a is known to lie within an ellipsoid:

$$a \in \mathcal{E} = \{\bar{a} + Pu \mid \|u\|_2 \leq 1\} \tag{65}$$

where $\bar{a} \in \mathbb{R}^n$ is the center of this ellipsoid and $P \in \mathbb{R}^{n \times n}$ is a positive semidefinite matrix. If generator i is our wind farm with 0 MW to 10 MW capacity, then $\bar{a}_i = -5$. If generator j is our solar farm with 0 MW to 20 MW capacity, then $\bar{a}_j = -10$. If generator k is our conventional natural gas plant with fixed 50 MW capacity, then $\bar{a}_k = -50$.

Matrix P maps the unit ball $\|u\|_2 \leq 1$ into an ellipse. Recall that $\lambda(P)$ provides the semi-axis lengths. Then we can define P to be a diagonal matrix with its eigenvalues on the diagonal. For example, the wind farm with 0 MW to 10 MW capacity would mean $P_{ii} = 5$. The solar farm with 0 MW to 20 MW capacity would mean $P_{jj} = 10$. The conventional natural gas plant with fixed 50 MW capacity would mean $P_{kk} = 0$. In this case, P is positive semi-definite. In a 100% renewable grid, then P would be strictly positive definite.

The robust version of (59)-(60) requires us to satisfy demand in EVERY instance of $a \in \mathcal{E}$:

$$\begin{aligned} \text{minimize} \quad & f^T x && (66)\\ \text{subject to} \quad & a^T x \leq b, \quad \forall\, a \in \mathcal{E} && (67)\\ & 0 \leq x \leq 1 && (68) \end{aligned}$$

Namely, find the optimal economic dispatch which satisfied demand D, given that real-time capacity vector a can exist anywhere in ellipsoid $\mathcal{E}$. An alternative description is the following. Find the optimal economic dispatch under the worst case scenario of real-time capacity vector $a \in \mathcal{E}$. Mathematically, we can reformulate the robust linear constraint (67) to examine the worst case:

$$\max\left\{a^T x \mid a \in \mathcal{E}\right\} \leq b \tag{69}$$

That is, we ensure adequate power generation in the face of uncertain renewable generation by ensuring (67) is satisfied in the absolute worst case. We can re-write the left hand side of (69) as

$$\begin{aligned} \max\left\{a^T x \mid a \in \mathcal{E}\right\} &= \bar{a}^T x + \max\left\{u^T P^T x \mid \|u\|_2 \leq 1\right\} && (70) \\ &= \bar{a}^T x + \|P^T x\|_2 && (71) \end{aligned}$$

Then the robust linear constraint can be re-expressed as

$$\bar{a}^T x + \|P^T x\|_2 \leq b \tag{72}$$

which is a second order cone constraint. Consequently, a robust LP can be converted to a second order cone program (SOCP) – a sub-class of convex optimization problems:

$$\begin{aligned} \text{minimize} \quad & f^T x && (73) \\ \text{subject to} \quad & \bar{a}^T x + \|P^T x\|_2 \leq b && (74) \\ & 0 \leq x \leq 1 && (75) \end{aligned}$$

Note that the additional norm term acts as a *regularization term*. Namely, it prevents x from being large in directions with considerable uncertainty.

Example 3.5 (Stochastic Dispatch in a High-Penetration Renewable Grid)**.** In the previous example, we optimize generator dispatch considering the *worst-case* scenario. One might argue this is too restrictive. That is, instead of ensuring feasibility for all possible real-time renewable generation capacity, we could potentially allow violations of (60). These are called “chance constraints.” In the economic dispatch application, a separate mechanism could be employed to balance electricity supply and demand, e.g. demand response, ancillary services, energy storage, etc.

To formalize this approach, recall the LP

$$\begin{aligned} \text{minimize} \quad & c^T x && (76) \\ \text{subject to} \quad & a^T x \leq b && (77) \end{aligned}$$

Assume that $a \in \mathbb{R}^n$ is Gaussian, i.e. $a \sim \mathcal{N}(\bar{a}, \Sigma)$. Then $a^T x$ is a Gaussian random variable with mean $\bar{a}^T x$ and variance $x^T \Sigma x$. Hence, we can express the probability that $a^T x \leq b$ is satisfied as

$$\Pr(a^T x \leq b) = \Phi\left(\frac{b - \bar{a}^T x}{\|\Sigma^{1/2} x\|_2}\right) \tag{78}$$

where $\Phi(x) = (1/\sqrt{2\pi}) \int_{-\infty}^{x} e^{-y^2/2} dy$ is the CDF of $\mathcal{N}(0,1)$. This enables us to relax (77) into

$$\begin{aligned} \text{minimize} \quad & c^T x && (79) \\ \text{subject to} \quad & \Pr(a^T x \leq b) \geq \eta && (80) \end{aligned}$$

The relaxed constraint (80) is called a *chance constraint*. Think of it as a reliability prescription. Namely, we require that $a^T x \leq b$ with a reliability of η, where η is typically 0.9, 0.95, or 0.99. Interestingly, we can use (78) to convert this stochastic LP into an SOCP as follows.

$$\begin{aligned} \text{minimize} \quad & c^T x && (81) \\ \text{subject to} \quad & \bar{a}^T x + \Phi^{-1}(\eta) \|\Sigma^{1/2} x\|_2 \leq b && (82) \end{aligned}$$

where $\Phi^{-1}(\cdot)$ is the inverse CDF for the Gaussian distribution. Note that we need $\Phi^{-1}(\cdot) \geq 0$ to be a valid second order cone constraint. This is true if and only if $\eta \geq 1/2$. Conveniently, we always want a reliability of $\eta \geq 1/2$ in practice.

To summarize, we can formulate a relaxed version of the robust LP problem using chance constraints. If the random variable a has a Gaussian distribution, then this chance constrained LP can be successfully re-formulated into a SOCP, which can be efficiently solved with convex programming solvers.

3.6 Maximum Likelihood Estimation

Imagine someone provides you with m data points for random variable y. You seek to fit a probability distribution to this data. In energy systems, variable y could represent wind speed, solar insolation, vehicle speed, building electricity demand, etc.

We consider a probability density function $p(y;\theta)$ for random variable y. Vector $\theta \in \mathbb{R}^n$ parameterizes this density. When $p(y;\theta)$ is considered as a function of θ for fixed y, then we call this a "likelihood function". As we shall see, it is extremely convenient to use the logarithm of this function, which we call the "log-likelihood function", denoted:

$$l(\theta) = \log p(y;\theta) \tag{83}$$

Let us return to the problem of fitting a probability distribution to this data. A standard method, called "maximum likelihood estimation (MLE)", is to estimate θ as

$$\hat{\theta} = \arg\max_{\theta}\ p(y;\theta) = \arg\max_{\theta} l(\theta) \tag{84}$$

for a given data point y. In this example, the optimization variable is θ. Symbol y is problem data. Interestingly, the MLE problem yields convex optimization problems in many common scenarios.

Specifically, (84) is a convex optimization problem if $l(\theta)$ is concave w.r.t. θ for each value of y. One can optimally add constraints that form a convex set as well.

Example 3.6 (MLE for Linear Models)**.** For concreteness, we consider a linear measurement model,

$$y_i = \theta^T \phi_i + v_i, \qquad i = 1, \cdots, m \tag{85}$$

where $\theta \in \mathbb{R}^n$ is the vector of parameters to be estimated, $y_i \in \mathbb{R}$ are the measured data points, $\phi_i \in \mathbb{R}^n$ are the regressors, and $v_i \in \mathbb{R}$ are the measurement errors or noise. Assume that v_i are independent and identically distributed (IID), with density $p(\cdot)$. The likelihood function, given the y_i's, is given by the products of the likelihood given each measurement y_i and regressor ϕ_i.

$$p(y; \theta) = \prod_{i=1}^{m} p(y_i - \theta^T \phi_i) \tag{86}$$

The log-likelihood function is then

$$l(\theta) = \log p(y; \theta) = \sum_{i=1}^{m} \log p(y_i - \theta^T \phi_i) \tag{87}$$

HINT: Recall the logarithm of a product property: $\log(ab) = \log(a) + \log(b)$. The MLE problem is:

$$\text{maximize} \quad \sum_{i=1}^{m} \log p(y_i - \theta^T \phi_i) \tag{88}$$

w.r.t. variable θ. As mentioned before, $p(y; \theta)$ is log-concave for several common probability distributions. For example, suppose that v_i are Gaussian with zero mean and variance σ^2. Thus $p(v) = (2\pi\sigma^2)^{-1/2} \cdot e^{-v^2/(2\sigma^2)}$. Substituting this expression for $p(\cdot)$ into (87) gives the log-likelihood function

$$l(x) = -\frac{m}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \|\Phi^T \theta - y\|_2^2 \tag{89}$$

where $\Phi = [\phi_1, \cdots, \phi_m] \in \mathbb{R}^{n \times m}$ is a concatenated matrix of regressor vectors. Consequently, we have shown the MLE problem for a Gaussian distribution is – quite simply – the solution to the least-squares problem

$$\hat{\theta} = \arg\min_{\theta} \|\Phi^T \theta - y\|_2^2 \tag{90}$$

How elegant!

Exercise 6. *Derive the MLE optimization formulation for (85) for the following distributions for v_i:*

1. *Laplacian noise distribution:* $p(v) = 1/(2a) \cdot e^{-|v|/a}$
2. *Uniform noise distribution:* $p(v) = 1/(2a)$ *on* $[-a, +a]$ *and zero elsewhere*

Example 3.7 (Logistic Regression)**.**

Example 3.8 (Mechanical Design w/ Geometric Programs).

3.7 Duality

4 Nonlinear Programming (NLP)

Nonlinear programming problems involve objective functions that are nonlinear in the decision variable x. LP and QP problems are special cases of NLPs. As such, the particular structure of LPs and QPs can be exploited for analysis and computation. In this section, we discuss a more general class of nonlinear problems and corresponding tools for analysis and computation.

A *nonlinear optimization problem* has the form

$$\min_{x} \quad f(x) \tag{91}$$
$$\text{subject to} \quad g_i(x) \leq 0, \quad i = 1, \cdots, m \tag{92}$$
$$h_j(x) = 0, \quad j = 1, \cdots, l. \tag{93}$$

Note the key difference is the objective function and constraints take a general form that is nonlinear in x. In this general case, we first discuss algorithms for the *unconstrained case*. Then we consider constraints and present general theory on NLPs.

4.1 Gradient Descent

Gradient descent is a first-order iterative algorithm for finding the local minimum of a differentiable function. It is applicable to unconstrained minimization problems. Starting from an initial guess, the main idea is to step in the direction of steepest descent at each iteration. Eventually the algorithm will converge when the gradient is zero, which corresponds to a local minimum.

This concept is illustrated in Fig. 17, which provides iso-contours of a function $f(x)$ that we seek to minimize. In this example, the user provides an initial guess x_0. Then the algorithm proceeds according to

$$x_{k+1} = x_k - h \cdot \nabla f(x) \tag{94}$$

where $h > 0$ is some positive step size. The iteration proceeds until a stopping criterion is satisfied. Typically, we stop when the gradient is sufficiently close to zero

$$\|\nabla f(x_k)\| \leq \epsilon \tag{95}$$

where $\epsilon > 0$ is some small user defined stopping criterion parameter, and the norm $\|\nabla f(x_k)\|$ can be a user-selected norm, such as the 2-norm, 1-norm, or ∞-norm.

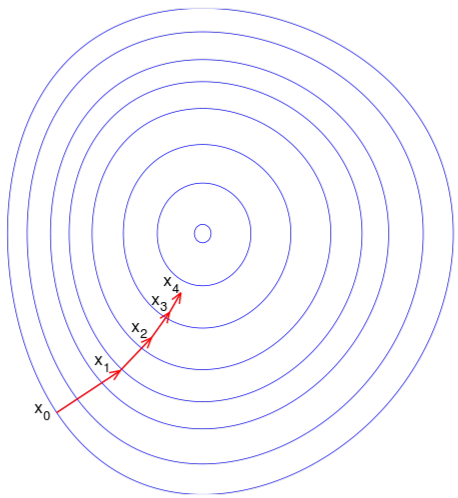


Figure 17: Illustration of gradient descent with step size proportional to the gradient.

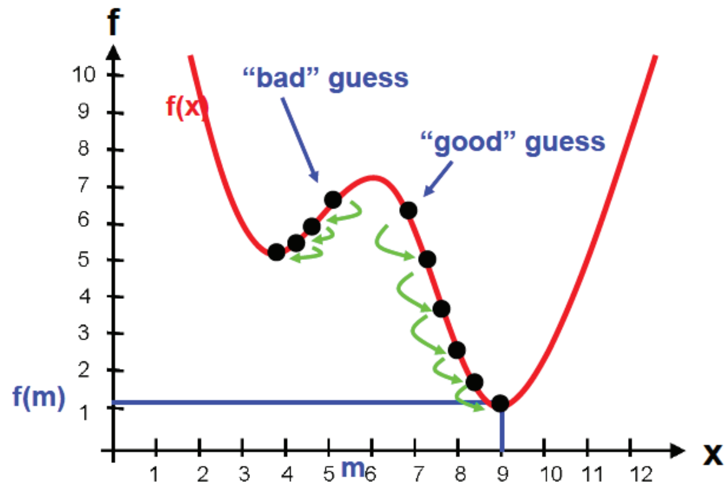


Figure 18: In non-convex functions, gradient descent converges to the local minimum. Consequently, different initial guesses may result in different solutions.

Exercise 7. *Minimize the function* $f(x_1, x_2) = \frac{1}{2}(x_1^2 + 10x_2^2)$ *with an initial guess of* $(x_{1,0}, x_{2,0}) = (10, 1)$. *Use a step-size of* $h = 1$, *and a stopping criterion of* $\|\nabla f(x_k)\|_2 = \sqrt{x_{1,k}^2 + x_{2,k}^2} \leq \epsilon = 0.01$.

For non-convex problems, such as the one illustrated in Fig. 18, the gradient descent algorithm converges to the local minimum. In other words, convergence to a global minimum is not guaranteed unless the function $f(x)$ is convex over the feasible set D. In this case, one may select a variety of initial guesses, x_0, to start the gradient descent algorithm. Then the best of all converged values is used for the proposed solution. This still does not guarantee a global minimum, but is effective at finding a sub-optimal solution in practice.

4.2 Barrier and Penalty Functions

A drawback of the gradient descent method is that it does not explicitly account for constraints. Barrier and penalty functions are two methods of augmenting the objective function $f(x)$ to approximately account for the constraints. To illustrate, consider the constrained minimization problem

$$\min_{x} \quad f(x) \tag{96}$$
$$\text{subject to} \quad g(x) \leq 0. \tag{97}$$

We seek to modify the objective function to account for the constraints, in an approximate way. Thus we can write

$$\min_{x} \quad f(x) + \phi(x; \epsilon) \tag{98}$$

where $\phi(x;\varepsilon)$ captures the effect of the constraints and is differentiable, thereby enabling usage of gradient descent. The parameter ε is a user-defined parameter that allows one to more accurately or more coarsely approximate the constraints. Barrier and penalty functions are two methods of defining $\phi(x;\varepsilon)$. The main idea of each is as follows:

- **Barrier Function:** Allow the objective function to increase towards infinity as x approaches the constraint boundary from inside the feasible set. In this case, the constraints are guaranteed to be satisfied, but it is impossible to obtain a boundary optimum.
- **Penalty Function:** Allow the objective function to increase towards infinity as x violates the constraints $g(x)$. In this case, the constraints can be violated, but it allows boundary optimum.

To motivate these methods, consider the non-convex function shown in Fig. 19. We seek to find the minimum within the range $[0.5, 1.5]$. Mathematically, this is a one-dimensional problem written as

$$\min_{x} \quad f(x) \tag{99}$$
$$\text{s. to} \quad x \leq b \tag{100}$$
$$x \geq a \tag{101}$$

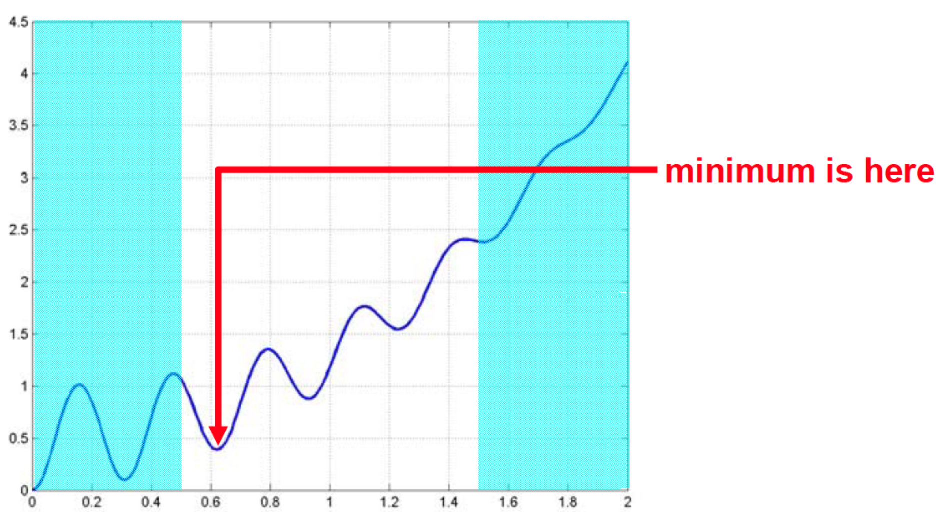


Figure 19: Find the optimum of the function shown above within the range [0.5, 1.5].

4.2.1 Log Barrier Function

Let us define the **log barrier function** as

$$\phi(x;\varepsilon) = -\varepsilon \log\left(\frac{(x-a)(b-x)}{b-a}\right) \tag{102}$$

The critical property of the log barrier function is that $\phi(x;\varepsilon) \to +\infty$ as $x \to a$ from the right side and $x \to b$ from the left side. Ideally, the log barrier function is zero inside the constraint set. This desired property becomes increasingly true as $\varepsilon \to 0$.

4.2.2 Quadratic Penalty Function

Let us define the **quadratic penalty function** as

$$\phi(x;\varepsilon) = \begin{cases} 0 & \text{if } a \le x \le b \\ \frac{1}{2\varepsilon}(x-a)^2 & \text{if } x < a \\ \frac{1}{2\varepsilon}(x-b)^2 & \text{if } x > b \end{cases} \tag{103}$$

The critical property of the quadratic penalty function is that $\phi(x;\varepsilon)$ increases towards infinity as x increases beyond b or decreases beyond a. The severity of this increase is parameterized by ε. Also, note that $\phi(x;\varepsilon)$ is defined such that $f(x) + \phi(x;\varepsilon)$ remains differentiable at $x = a$ and $x = b$, thus enabling application of the gradient descent algorithm.

4.3 Sequential Quadratic Programming (SQP)

In our discussion of NLPs so far, we have explained how to solve (i) unconstrained problems via gradient method, and (ii) unconstrained problems augmented with barrier or penalty functions to account for constraints. In this section, we provide a direct method for handling NLPs with constraints, called the Sequential Quadratic Programming (SQP) method. The idea is simple. We solve a single NLP as a sequence QP subproblems. In particular, at each iteration we approximate the objective function and constraints by a QP. Then, within each iteration, we solve the corresponding QP and use the solution as the next iterate. This process continues until an appropriate stopping criterion is satisfied.

SQP is very widely used in engineering problems and often the first “go-to” method for NLPs. For many practical energy system problems, it produces fast convergence thanks to its strong theoretical basis. This method is commonly used under-the-hood of Matlab function `fmincon`.

Consider the general NLP

$$\min_{x} \qquad f(x) \tag{104}$$

$$\text{subject to} \quad g(x) \leq 0, \tag{105}$$

$$h(x) = 0, \tag{106}$$

and the k^{th} iterate x_k for the decision variable. We utilize the Taylor series expansion. At each iteration of SQP, we consider the 2nd-order Taylor series expansion of the objective function (104), and 1st-order expansion of the constraints (105)-(106) around $x = x_k$:

$$f(x) \approx f(x_k) + \frac{\partial f}{\partial x}^T (x_k)\,(x - x_k) + \frac{1}{2}\,(x - x_k)^T\,\frac{\partial^2 f}{\partial x^2}(x_k)\,(x - x_k)\,, \tag{107}$$

$$g(x) \approx g(x_k) + \frac{\partial g}{\partial x}^T (x_k)\,(x - x_k) \leq 0, \tag{108}$$

$$h(x) \approx h(x_k) + \frac{\partial h}{\partial x}^T (x_k)\,(x - x_k) = 0 \tag{109}$$

To simplify the notation, define $\tilde{x} = x - x_k$. Then we arrive at the following approximate QP

$$\min \quad \frac{1}{2}\tilde{x}^T Q \tilde{x} + R^T \tilde{x} \tag{110}$$

$$\text{s. to} \quad A\tilde{x} \leq b \tag{111}$$

$$A_{eq}\tilde{x} = b_{eq} \tag{112}$$

where

$$Q = \frac{\partial^2 f}{\partial x^2}(x_k), \qquad R = \frac{\partial f}{\partial x}(x_k) \tag{113}$$

$$A = \frac{\partial g}{\partial x}^T(x_k), \qquad b = -g(x_k) \tag{114}$$

$$A_{eq} = \frac{\partial h}{\partial x}^T(x_k), \qquad b_{eq} = -h(x_k) \tag{115}$$

Suppose (110)-(112) yields the optimal solution $\tilde{x}^*$. Then let $x_{k+1} = x_k + \tilde{x}^*$, and repeat.

Remark 4.1. Note that the iterates in SQP are not guaranteed to be feasible for the original NLP problem. That is, it is possible to obtain a solution to the QP subproblem which satisfies the approximate QP's constraints, but not the original NLP constraints.

Example 4.1. *Consider the NLP*

$$\min_{x_1,x_2} \quad e^{-x_1} + (x_2 - 2)^2 \tag{116}$$

$$\textit{s. to} \quad x_1 x_2 \leq 1. \tag{117}$$

with the initial guess $[x_{1,0}, x_{2,0}]^T = [1, 1]^T$*. By hand, formulate the Q,R,A,b matrices for the first three iterates. Use Matlab command* `quadprog` *to solve each subproblem. What is the solution after three iterations?*

We have $f(x) = e^{-x_1} + (x_2 - 2)^2$ and $g(x) = x_1 x_2 - 1$. The iso-contours for the objective function and constraint are provided in Fig. 20. From visual inspection, it is clear the optimal solution is near $[0.5, 2]^T$. We seek to find the approximate QP subproblem

$$\min \quad \frac{1}{2}\tilde{x}^T Q \tilde{x} + R^T \tilde{x} \tag{118}$$

$$\text{s. to} \quad A\tilde{x} \leq b \tag{119}$$

Taking derivatives of $f(x)$ and $g(x)$, one obtains

$$Q = \begin{bmatrix} e^{-x_1} & 0 \\ 0 & 2 \end{bmatrix}, \qquad R = \begin{bmatrix} -e^{-x_1} \\ 2(x_2 - 2) \end{bmatrix}, \tag{120}$$

$$A = \begin{bmatrix} x_2 & x_1 \end{bmatrix}, \qquad b = 1 - x_1 x_2 \tag{121}$$

Now consider the initial guess $[x_{1,0}, x_{2,0}]^T = [1, 1]^T$. Note that this guess is feasible. We obtain the following matrices for the first QP subproblem

$$Q = \begin{bmatrix} e^{-1} & 0 \\ 0 & 2 \end{bmatrix}, \qquad R = \begin{bmatrix} -e^{-1} \\ -2 \end{bmatrix},$$

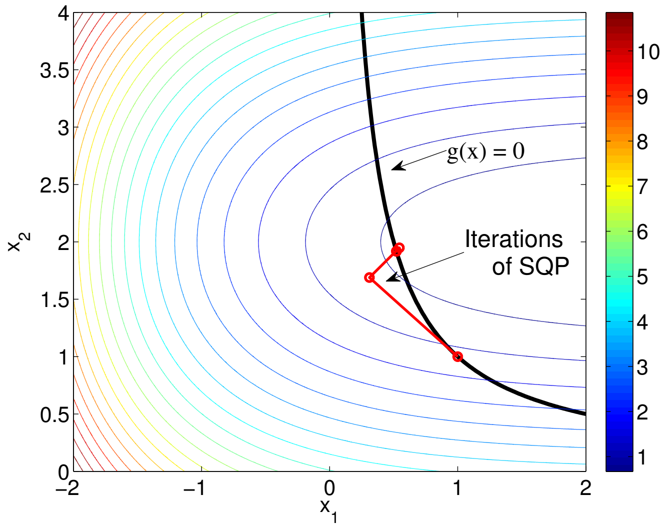
$$A = \begin{bmatrix} 1 & 1 \end{bmatrix}, \qquad b = 0$$

Solving this QP subproblem results in $\tilde{x}^* = [-0.6893, 0.6893]$. Then the next iterate is given by $[x_{1,1}, x_{2,1}] = [x_{1,0}, x_{2,0}] + \tilde{x}^* = [0.3107, 1.6893]$. Repeating the formulation and solution of the QP subproblem at iteration 1 produces $[x_{1,1}, x_{2,1}] = [0.5443, 1.9483]$. Note that this iterate is infeasible. Continued repetitions will produce iterates that converge toward the true solution.

SQP provides an algorithmic way to solve NLPs in energy system applications. However, it still relies on approximations - namely truncated Taylor series expansions - to solve the optimization problem via a sequence of QP subproblems. Next, we discuss a direct method for solving NLPs, without approximation.

4.4 First-Order Necessary Conditions for Optimality

In calculus, you learned that a necessary condition for minimizers is that the function's slope is zero at the minimum. We extend this notion in this section. Namely, we discuss first-order necessary conditions for optimality for NLPs.



Iter.	$[x_1, x_2]$	$f(x)$	$g(x)$
0	$[1, 1]$	1.3679	0
1	$[0.311, 1.689]$	0.8295	-0.4751
2	$[0.544, 1.948]$	0.5829	0.0605
3	$[0.522, 1.917]$	0.6002	0.0001
4	$[0.521, 1.919]$	0.6004	-0.0000

Figure 20 & Table 1: [LEFT] Iso-contours of objective function and constraint for Example 4.1. [RIGHT] Numerical results for first three iterations of SQP. Note that some iterates are infeasible.

4.4.1 Method of Lagrange Multipliers

Consider the equality constrained optimization problem

$$\min \quad f(x) \tag{122}$$

$$\text{s. to} \quad h_j(x) = 0, \qquad j = 1, \cdots, l \tag{123}$$

Introduce the so-called "Lagrange multipliers" $\lambda_j, j = 1, \cdots, l$. Then we can augment the cost function to form the "Lagrangian" $L(x)$ as follows

$$L(x) = f(x) + \sum_{j=1}^{l} \lambda_j h_j(x) \tag{124}$$

$$= f(x) + \lambda^T h(x) \tag{125}$$

Note that when all constraints are satisfied, that is $h(x) = 0$, then the second term becomes zero. Consequently, the Lagrangian $L(x)$ and cost function $f(x)$ provide identical values for all feasible x. We now state the first-order necessary condition (FONC) for equality constrained problems:

Proposition 3 (FONC for Equality Constrained NLPs). *If a local minimum x^* exists, then it satisfies*

$$\frac{\partial L}{\partial x}(x^*) = \frac{\partial f}{\partial x}(x^*) + \lambda^T \frac{\partial h}{\partial x}(x^*) = 0 \qquad \textit{(stationarity)}, \tag{126}$$

$$\frac{\partial L}{\partial \lambda}(x^*) = h(x^*) = 0 \qquad \textit{(feasibility)}. \tag{127}$$

That is, the gradient of the Lagrangian is zero at the minimum x^*.

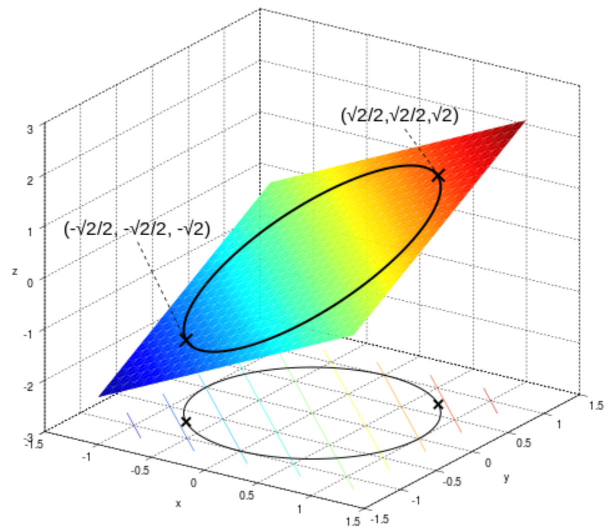


Figure 21: Visualization of circle-plane problem from Example 4.4.

Remark 4.2. *This condition is only* <u>*necessary*</u>*. That is, if a local minimum* x^* *exists, then it must satisfy the FONC. However, a design* x *which satisfies the FONC isn't necessarily a local minimum.*

Remark 4.3. *If the optimization problem is* <u>*convex*</u>*, then the FONC is* <u>*necessary and sufficient*</u>*. That is, a design* x *which satisfies the FONC is also a local minimum.*

Example 4.2. Consider the equality constrained QP

$$\min \quad \frac{1}{2}x^TQx + R^Tx \tag{128}$$

$$\text{s. to} \quad Ax = b \tag{129}$$

Form the Lagrangian,

$$L(x) = \frac{1}{2}x^TQx + R^Tx + \lambda^T(Ax - b). \tag{130}$$

Then the FONC is

$$\frac{\partial L}{\partial x}(x^*) = Qx^* + R + A^T\lambda = 0. \tag{131}$$

Combining the FONC with the equality constraint yields

$$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \lambda \end{bmatrix} = \begin{bmatrix} -R \\ b \end{bmatrix} \tag{132}$$

which provides a set of linear equations that can be solved directly.

Example 4.3. Consider a circle inscribed on a plane, as shown in Fig. 21. Suppose we wish to find the "lowest" point on the plane while being constrained to the circle. This can be abstracted as the NLP:

$$\begin{aligned} \text{min} \quad & f(x,y) = x + y && (133) \\ \text{s. to} \quad & x^2 + y^2 = 1 && (134) \end{aligned}$$

Form the Lagrangian

$$L(x,y,\lambda) = x + y + \lambda(x^2 + y^2 - 1) \tag{135}$$

Then the FONCs and equality constraint can be written as the set of nonlinear equations:

$$\begin{aligned} \frac{\partial L}{\partial x} &= 1 + 2\lambda x = 0 && (136) \\ \frac{\partial L}{\partial y} &= 1 + 2\lambda y = 0 && (137) \\ \frac{\partial L}{\partial \lambda} &= x^2 + y^2 - 1 = 0 && (138) \end{aligned}$$

One can solve these three equations for x, y, λ by hand to arrive at the solution

$$\begin{aligned} (x^*, y^*) &= \left(\pm\frac{\sqrt{2}}{2}, \pm\frac{\sqrt{2}}{2} \right) \\ f(x^*, y^*) &= \pm\sqrt{2} \\ \lambda &= \mp 1/\sqrt{2} \end{aligned}$$

4.4.2 Karush-Kuhn-Tucker (KKT) Conditions

Now we consider the general constrained optimization problem

$$\begin{aligned} \text{min} \quad & f(x) && (139) \\ \text{s. to} \quad & g_i(x) \leq 0, \qquad i = 1, \cdots, m && (140) \\ & h_j(x) = 0, \qquad j = 1, \cdots, l && (141) \end{aligned}$$

Introduce the so-called "Lagrange multipliers" $\lambda_j, j = 1, \cdots, l$ each associated with equality constraints $h_j(x), j = 1, \cdots, l$ and $\mu_i, i = 1, \cdots, m$ each associated with inequality constraints $g_i(x), i = 1, \cdots, m$. Then we can augment the cost function to form the "Lagrangian" $L(x)$ as follows

$$\begin{aligned} L(x) &= f(x) + \sum_{i=1}^{m} \mu_i g_i(x) + \sum_{j=1}^{l} \lambda_j h_j(x) && (142) \\ &= f(x) + \mu^T g(x) + \lambda^T h(x) && (143) \end{aligned}$$

As before, when the equality constraints are satisfied, $h(x) = 0$, then the third term becomes zero. Elements of the second term become zero in two cases: (i) an inequality constraint is active, that is $g_i(x) = 0$; (ii) the Lagrange multiplier $\mu_i = 0$. Consequently, the Lagrangian $L(x)$ can be constructed to have identical values of the cost function $f(x)$ if the aforementioned conditions are applied. This motivates the first-order necessary conditions (FONC) for the general constrained optimization problem – called the Karush-Kuhn-Tucker (KKT) Conditions.

Proposition 4 (KKT Conditions)**.** *If x^* is a local minimum, then the following necessary conditions hold:*

$$\frac{\partial f}{\partial x}(x^*) + \sum_{i=1}^{m} \mu_i \frac{\partial}{\partial x} g_i(x^*) + \sum_{j=1}^{l} \lambda_j \frac{\partial}{\partial x} h_j(x^*) = 0, \qquad \textit{Stationarity} \tag{144}$$

$$\begin{aligned} g_i(x^*) &\leq 0, \quad i = 1, \cdots, m \qquad \textit{Feasibility} && (145) \\ h_j(x^*) &= 0, \quad j = 1, \cdots, l \qquad \textit{Feasibility} && (146) \\ \mu_i &\geq 0, \quad i = 1, \cdots, m \qquad \textit{Non-negativity} && (147) \\ \mu_i g_i(x^*) &= 0, \quad i = 1, \cdots, m \qquad \textit{Complementary slackness} && (148) \end{aligned}$$

which can also be written in matrix-vector form as

$$\frac{\partial f}{\partial x}(x^*) + \mu^T \frac{\partial}{\partial x} g(x^*) + \lambda^T \frac{\partial}{\partial x} h(x^*) = 0, \qquad \textit{Stationarity} \tag{149}$$

$$\begin{aligned} g(x^*) &\leq 0, \qquad \textit{Feasibility} && (150) \\ h(x^*) &= 0, \qquad \textit{Feasibility} && (151) \\ \mu &\geq 0, \qquad \textit{Non-negativity} && (152) \\ \mu^T g(x^*) &= 0, \qquad \textit{Complementary slackness} && (153) \end{aligned}$$

Remark 4.4. Note the following properties of the KKT conditions

- Non-zero μ_i indicates $g_i \leq 0$ is active (true with equality). In practice, non-zero μ_i is how we identify active constraints from nonlinear solvers.
- The KKT conditions are necessary, only. That is, if a local minimum x^* exists, then it must satisfy the KKT conditions. However, a design x which satisfies the KKT conditions isn't necessarily a local minimum.
- If problem is convex, then the KKT conditions are necessary and sufficient. That is, one may directly solve the KKT conditions to obtain the minimum.
- Lagrange multipliers λ, μ are sensitivities to perturbations in the constraints

  - In economics, this is called the "shadow price"
  - In control theory, this is called the "co-state"

- The KKT conditions have a geometric interpretation demonstrated in Fig. 22. Consider minimizing the cost function with isolines shown in red, where $f(x)$ is increasing as x_1, x_2 increase, as shown by the gradient vector ∇f. Now consider two inequality constraints $g_1(x) \leq 0, g_2(x) \leq 0$, forming the feasible set colored in light blue. The gradients at the minimum, weighted by the Lagrange multipliers, are such that their sum equals $-\nabla f$. In other words, the vectors balance to zero according to $\nabla f(x^*) + \mu_1 \nabla g_1(x^*) + \mu_2 \nabla g_2(x^*) = 0$.

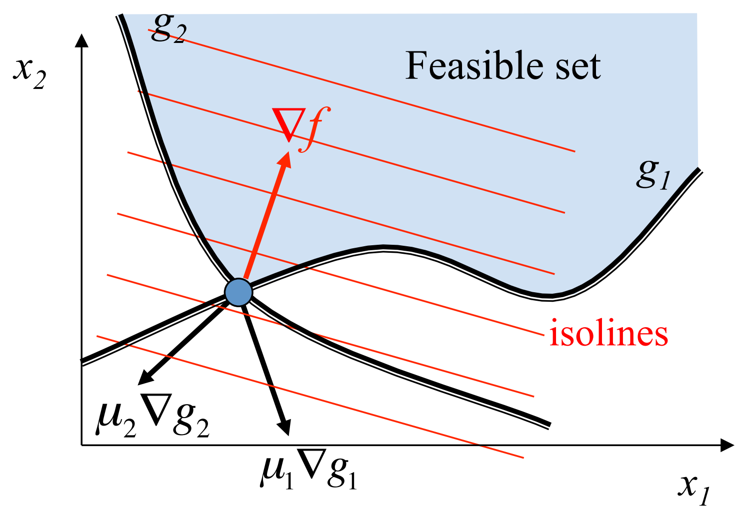


Figure 22: Geometric interpretation of KKT conditions

Example 4.4. Consider again the circle-plane problem, as shown in Fig. 21. Suppose we wish to find the "lowest" point on the plane while being constrained to within or on the circle. This can be abstracted as the NLP:

$$\min \quad f(x, y) = x + y \tag{154}$$

$$\text{s. to} \quad x^2 + y^2 \leq 1 \tag{155}$$

Note this problem is convex, therefore the solution to the KKT conditions provides the minimizer (x^*, y^*). We form the Lagrangian

$$L(x, y, \mu) = x + y + \mu(x^2 + y^2 - 1) \tag{156}$$

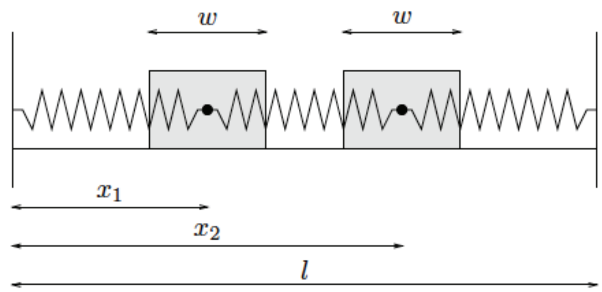


Figure 23: Spring-block system for Example 4.5

Then the KKT conditions are

$$\frac{\partial L}{\partial x} = 1 + 2\mu x^* = 0 \tag{157}$$
$$\frac{\partial L}{\partial y} = 1 + 2\mu y^* = 0 \tag{158}$$
$$\frac{\partial L}{\partial \mu} = (x^*)^2 + (y^*)^2 - 1 \leq 0 \tag{159}$$
$$\mu \geq 0 \tag{160}$$
$$\mu\left((x^*)^2 + (y^*)^2 - 1\right) = 0 \tag{161}$$

One can solve these equations/inequalities for x^*, y^*, μ by hand to arrive at the solution

$$\begin{aligned} (x^*, y^*) &= \left(-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right) \\ f(x^*, y^*) &= -\sqrt{2} \\ \mu &= 1/\sqrt{2} \end{aligned}$$

Example 4.5 (Mechanics Interpretation)**.** Interestingly, the KKT conditions can be used to solve a familiar undergraduate physics example involving the principles of mechanics. Consider two blocks of width w, where each block is connected to each other and the surrounding walls by springs, as shown in Fig. 23. Reading left to right, the springs have spring constants k_1, k_2, k_3. The objective is to determine the equilibrium position of the masses. The principles of mechanics indicate that the equilibrium is achieved when the spring potential energy is minimized. Moreover, we have *kinematic constraints* that restrain the block positions. That is, the blocks cannot overlap with each other or the walls. Consequently, we can formulate the following nonlinear program.

$$\text{minimize} \quad f(x_1, x_2) = \frac{1}{2}k_1 x_1^2 + \frac{1}{2}k_2\left(x_2 - x_1\right)^2 + \frac{1}{2}k_3\left(l - x_2\right)^2 \tag{162}$$
$$\text{subject to} \quad x_1 - \frac{w}{2} \geq 0, \tag{163}$$

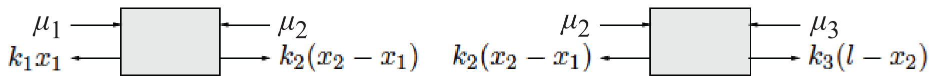


Figure 24: Free-body diagram of spring-block system for Example 4.5

$$x_1 + \frac{w}{2} \leq x_2 - \frac{w}{2}, \tag{164}$$
$$x_2 + \frac{w}{2} \leq l \tag{165}$$

It is easy to see this problem is a QP with a convex feasible set. Consequently, we may formulate and solve the KKT conditions directly to find the equilibrium block positions.

Consider Lagrange multipliers μ_1, μ_2, μ_3. Form the Lagrangian:

$$L(x, \mu) = \frac{1}{2}k_1 x_1^2 + \frac{1}{2}k_2 (x_2 - x_1)^2 + \frac{1}{2}k_3 (l - x_2)^2 + \mu_1 \left(\frac{w}{2} - x_1\right) + \mu_2 (x_1 - x_2 + w) + \mu_3 \left(x_2 + \frac{w}{2} - l\right) \tag{166}$$

where $x = [x_1, x_2]^T, \mu = [\mu_1, \mu_2, \mu_3]^T$. Now we can formulate the KKT conditions: We have $\mu \geq 0$ for non-negativity,

$$\mu_1 \left(\frac{w}{2} - x_1\right) = 0, \qquad \mu_2 (x_1 - x_2 + w) = 0, \qquad \mu_3 \left(x_2 + \frac{w}{2} - l\right) = 0 \tag{167}$$

for complementary slackness, and

$$\begin{bmatrix} k_1 x_1 - k_2 (x_2 - x_1) \\ k_2 (x_2 - x_1) - k_3 (l - x_2) \end{bmatrix} + \mu_1 \begin{bmatrix} -1 \\ 0 \end{bmatrix} + \mu_2 \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \mu_3 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0 \tag{168}$$

for stationarity. Interestingly, the μ_i's can be interpreted as contact forces. That is, consider the free-body diagrams for each block shown in Fig. 24, where we denote the contact forces between the left wall–block 1, block 1–block 2, and block 2–right wall for μ_1, μ_2, μ_3, respectively. When no contact exists, then the corresponding contact force is trivially zero, which also indicates the associated inequality constraint is inactive. However, when the contact force μ_i is non-zero, this indicates the corresponding inequality constraint is active.

References

[1] P. Denholm, E. Ela, B. Kirby, and M. Milligan, “Role of energy storage with renewable electricity generation,” National Renewable Energy Laboratory (NREL), Golden, CO., Tech. Rep. Technical Report NREL/TP-6A2-47187, 2010.

[2] “Strategic design of public bicycle sharing systems with service level constraints,” Transportation Research Part E: Logistics and Transportation Review, vol. 47, no. 2, pp. 284 – 294, 2011.

[3] J. C. Garcia-Palomares, J. Gutierrez, and M. Latorre, "Optimizing the location of stations in bike-sharing programs: A gis approach," Applied Geography, vol. 35, no. 1-2, pp. 235–246, 2012.

[4] R. Nair, E. Miller-Hooks, R. C. Hampshire, and A. Busic, "Large-scale vehicle sharing systems: Analysis of velib," International Journal of Sustainable Transportation, vol. 7, no. 1, pp. 85–106, 2013.

[5] S. Bashash, S. J. Moura, J. C. Forman, and H. K. Fathy, "Plug-in hybrid electric vehicle charge pattern optimization for energy cost and battery longevity," Journal of Power Sources, vol. 196, no. 1, pp. 541 – 549, 2011.

[6] S. J. Tong, A. Same, M. A. Kootstra, and J. W. Park, "Off-grid photovoltaic vehicle charge using second life lithium batteries: An experimental and numerical investigation," Applied Energy, vol. 104, no. 0, pp. 740 – 750, 2013.

[7] P. Trnka, J. Pekar, and V. Havlena, "Application of distributed mpc to barcelona water distribution network," in Proceedings of the 18th IFAC World Congress. Milan, Italy, 2011.

[8] S. Boyd and L. Vandenberghe, Convex optimization. Cambridge university press, 2009.

5 Appendix: Linear Programming (LP) in Detail

We begin our exposition of linear programming problems with the following example.

Example 5.1 (Building a Solar Array Farm)**.** You are tasked with designing the parameters of a new photovoltaic array installation. Namely, you must decide on the square footage of the photovoltaic arrays, and the power capacity of the power electronics which interface the generators to the grid. The goal is to minimize installation costs, subject to the following constraints:

1. You cannot select negative PV array area, nor negative power electronics power capacity.
2. The minimum generating capacity for the photovoltaic array is $g_{\min}$.
3. The power capacity of the power electronics must be greater than or equal to the PV array power capacity.
4. The available spatial area for installation is limited by $s_{\max}$.
5. You have a maximum budget of $b_{\max}$.

Using the notation in Table 2, we can formulate the following optimization problem:

$$\min_{x_1,x_2} \quad c_1 x_1 + c_2 x_2 \tag{169}$$

$$\text{subject to:} \quad x_1 \geq 0 \tag{170}$$

$$x_2 \geq 0 \tag{171}$$

$$a_1 x_1 \geq g_{\min} \tag{172}$$

$$a_1 x_1 \leq x_2 \tag{173}$$

$$x_1 \leq s_{\max} \tag{174}$$

$$c_1 x_1 + c_2 x_2 \leq b_{\max} \tag{175}$$

Table 2: Building a Solar Array Farm

Spatial area of photovoltaic arrays [m^2]	x_1
Power capacity of power electronics [kW]	x_2
Cost of square meter of PV array [USD/m^2]	c_1
Cost of power electronics per kW [USD/kW]	c_2
Min PV array generating capacity [kW]	$g_{\min}$
Power of PV array per area [kW/m^2]	a_1
Max spatial area [m^2]	$s_{\max}$
Maximum budget [USD]	$b_{\max}$

where (170)-(171) encode constraint 1, and (172)-(175) respectively encode constraints 2-5. Rearranging all inequality constraints into less-than-or-equal-to, we arrive at the so-called standard form:

$$\begin{aligned}
\min_{x_1,x_2} \quad & c_1 x_1 + c_2 x_2 && (176)\\
\text{subject to:} \quad & -x_1 \leq 0 && (177)\\
& -x_2 \leq 0 && (178)\\
& -a_1 x_1 \leq -g_{\min} && (179)\\
& a_1 x_1 - x_2 \leq 0 && (180)\\
& x_1 \leq s_{\max} && (181)\\
& c_1 x_1 + c_2 x_2 \leq b_{\max} && (182)
\end{aligned}$$

which can be written into vector-matrix form as

$$\begin{aligned}
\min_{x} \quad & c^T x && (183)\\
\text{subject to:} \quad & Ax \leq b && (184)
\end{aligned}$$

where $x = [x_1, x_2]$ and

$$c = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}, \qquad A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ -a_1 & 0 \\ a_1 & -1 \\ 1 & 0 \\ c_1 & c_2 \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ 0 \\ -g_{\min} \\ 0 \\ s_{\max} \\ b_{\max} \end{bmatrix}. \tag{185}$$

Note that the feasible set (184) is a convex set, according to Remark 2.1. Also, the objective function (183) is linear and therefore convex. Consequently, this is a convex optimization problem. More specifically, this falls within a special subset of convex problems, called linear programs. The optimization problem (183)-(184) is known as a **linear program** (LP), and is fully characterized by matrices c, A, b. The key trait of a LP is that the objective function and constraints are linear functions of the design variables x. Now that we have successively formulated the Solar Array Farm problem as a LP, the problem is essentially solved. In Matlab, one may utilize the `linprog` command to solve the optimization problem, given matrices c, A, b.

Remark 5.1 (Program Reduction)**.** Sometimes constraints take the form of equalities. From Example 5.1, suppose the power capacity of the electronics must be 1.2 times the PV array generating capacity, i.e.

$$x_2 = 1.2\; a_1 x_1 \tag{186}$$

We can replace x_2 by the expression above and reduce the problem size:

$$\begin{aligned}
\min_{x_1} \quad & (c_1 + 1.2\; a_1 c_2)\, x_1 && && (187)\\
\text{subject to:} \quad & -x_1 &\leq\;& 0 && (188)\\
& -1.2\; a_1 x_1 &\leq\;& 0 && (189)\\
& -a_1 x_1 &\leq\;& -g_{\min} && (190)\\
& -0.2 a_1 x_1 &\leq\;& 0 && (191)\\
& x_1 &\leq\;& s_{\max} && (192)\\
& (c_1 + 1.2\; a_1 c_2)\, x_1 &\leq\;& b_{\max} && (193)
\end{aligned}$$

Note that some inequalities **dominate** others. For example, constraint (190) **dominates** (188), (189), (191) in the sense that (188), (189), (191) are automatically verified when constraint (190) is true. Thus we can further reduce the problem to

$$\begin{aligned}
\min_{x_1} \quad & (c_1 + 1.2\; a_1 c_2)\, x_1 && && (194)\\
\text{subject to:} \quad & -a_1 x_1 &\leq\;& -g_{\min} && (195)\\
& x_1 &\leq\;& s_{\max} && (196)\\
& (c_1 + 1.2\; a_1 c_2)\, x_1 &\leq\;& b_{\max} && (197)
\end{aligned}$$

We now formalize the notion of constraint domination with the following definition.

Definition 5.1 (Constraint Domination)**.** *Inequality constraint i <u>dominates</u> inequality constraint j when satisfaction of constraint j is automatically verified by constraint i.*

Continuing with our example, intuition suggests we can minimize costs by selecting the spatial area x_1 such that the minimum generating capacity is met exactly[1]. That is, at the optimal solution x_1^*, constraint (190) is true with equality. This produces the minimizer $x_1^* = g_{\min}/a_1$ and the minimum installation cost $(c_1 + 1.2\; a_1\; c_2)\, g_{\min}/a_1$. Note that we denote minimizers with the star notation: x_1^*. Also, when an inequality constraint is true with equality at the optimum, we call the constraint an **active constraint**.

[1]Naturally, this assumes the problem parameters are such that the feasible set is non-empty, that is, some solution exists.

Definition 5.2 (Active Inequalities)**.** *Constraint i is called an active constraint when it is true with equality at the optimum.*

5.1 General Form of LP

Linear programs can be identified as having linear objective functions and linear constraints. Suppose we have N decision variables x_i, M inequality constraints, and L equality constraints. Mathematically, this takes the form

$$\begin{aligned}
\min \quad & c_1 x_1 + c_2 x_2 + \ldots + c_N x_N \\
\text{subject to:} \quad & a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,N}x_N \leq b_1, \\
& a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,N}x_N \leq b_2, \\
& \vdots \\
& a_{M,1}x_1 + a_{M,2}x_2 + \ldots + a_{M,N}x_N \leq b_M, \\
& a_{eq,1,1}x_1 + a_{eq,1,2}x_2 + \ldots + a_{eq,1,N}x_N = b_{eq,1}, \\
& a_{eq,2,1}x_1 + a_{eq,2,2}x_2 + \ldots + a_{eq,2,N}x_N = b_{eq,2}, \\
& \vdots \\
& a_{eq,L,1}x_1 + a_{eq,L,2}x_2 + \ldots + a_{eq,L,N}x_N = b_{eq,L},
\end{aligned}$$

We may equivalently write this problem in "Sigma" notation as follows:

$$\begin{aligned}
\min \quad & \sum_{k=1}^{N} c_k x_k \\
\text{subject to:} \quad & \sum_{k=1}^{N} a_{1,k} x_k \leq b_1, \\
& \sum_{k=1}^{N} a_{2,k} x_k \leq b_2, \\
& \vdots \\
& \sum_{k=1}^{N} a_{M,k} x_k \leq b_M, \\
& \sum_{k=1}^{N} a_{eq,1,k} x_k = b_{eq,1}, \\
& \sum_{k=1}^{N} a_{eq,2,k} x_k = b_{eq,2}, \\
& \vdots \\
& \sum_{k=1}^{N} a_{eq,L,k} x_k = b_{eq,L}
\end{aligned}$$

The most compact notation uses matrix-vector format, and is given by

$$\begin{aligned} \min \quad & c^T x \\ \text{subject to:} \quad & Ax \leq b, \\ & A_{eq} x = b_{eq}, \end{aligned}$$

where

$$\begin{aligned} x &= [x_1, x_2, \ldots, x_N]^T \\ c &= [c_1, c_2, \ldots, c_N]^T \\ [A]_{i,j} &= a_{i,j}, \quad A \in \mathbb{R}^{M \times N} \\ b &= [b_1, b_2, \ldots, b_M]^T, \quad b \in \mathbb{R}^M \\ [A_{eq}]_{i,j} &= a_{eq,i,j}, \quad A_{eq} \in \mathbb{R}^{L \times N} \\ b_{eq} &= [b_{eq,1}, b_{eq,2}, \ldots, b_{eq,L}]^T, \quad b_{eq} \in \mathbb{R}^L \end{aligned}$$

5.2 Graphical LP

For problems of one, two, or sometimes three dimensions, we can use graphical methods to visualize the feasible sets and solutions. This visualization provides excellent intuition for the nature of LP solutions. We demonstrate with the following example.

$$\begin{aligned} \max \quad & Z = 140x_1 + 160x_2 \\ \\ \text{s. to} \quad & 2x_1 + 4x_2 \leq 28 \\ & 5x_1 + 5x_2 \leq 50 \\ & x_1 \leq 8 \\ & x_2 \leq 6 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{aligned}$$

On a graph, one may successively plot each of the inequality constraints and divide the Cartesian space into feasible "half-spaces." As each half-space is identified, we retain the intersection of the remaining feasible set. This successive construction provides the **feasible set**, as shown in Fig. 25. After constructing the feasible set, we can plot the iso-contours of the objective function. For example, the lower-left-most iso-contour in Fig. 25 corresponds to $Z = 0$. Continuing towards the upper-right, the value of the objective function increases. The intersection of the largest-valued iso-contour and the feasible set occurs when $Z^* = 1480$, at $x_1^* = 6, x_2^* = 4$. Consequently, we have graphically solved the LP.

A feasible set can fall within one of the following three categories:

- **[Bounded]** The feasible set is bounded if it forms a closed subset of the Cartesian plane that does not include infinity.

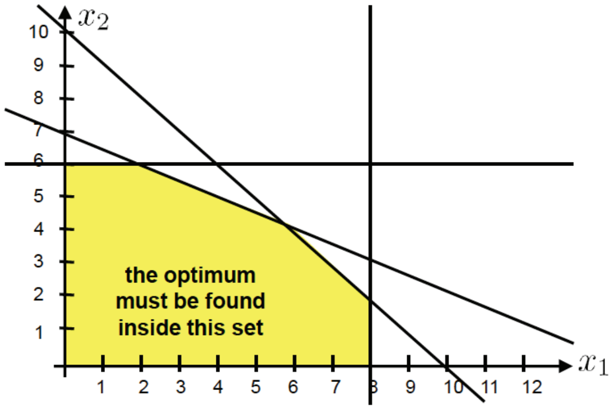


Figure 25: Construction of the feasible set in a linear program [LEFT], and the objective function isolines [RIGHT].

- **[Unbounded]** The feasible set is <u>unbounded</u> if it forms a subset of the Cartesian plane that includes infinity.
- **[Empty]** The feasible set is <u>empty</u> if the intersection of all the inequality constraints forms the empty set. In this case the problem is infeasible. That is, no solution exists.

Exercise 8. *Draw examples of each of the three categories given above.*

Also note that an objective function may be <u>bounded</u> or <u>unbounded</u>. We make these concepts concrete with the following examples. Consider the feasible set defined by inequalities $x_2 \geq 3$, $x_1 + x_2 \geq 6$, and $x_2 \leq 0.5x_1 + 7$, as shown in Fig. 26. On the left, consider the objective $\max Z = 3.24x_1$. The iso-contours continue towards $x_1 = \infty$, without being bounded by the feasible set. Hence, objective function Z is unbounded. In contrast, consider the objective $\min Z = x_1 + 3x_2$. Although the feasible set is unbounded, the iso-contours are bounded as they decrease in value. In this case, objective function Z is bounded.

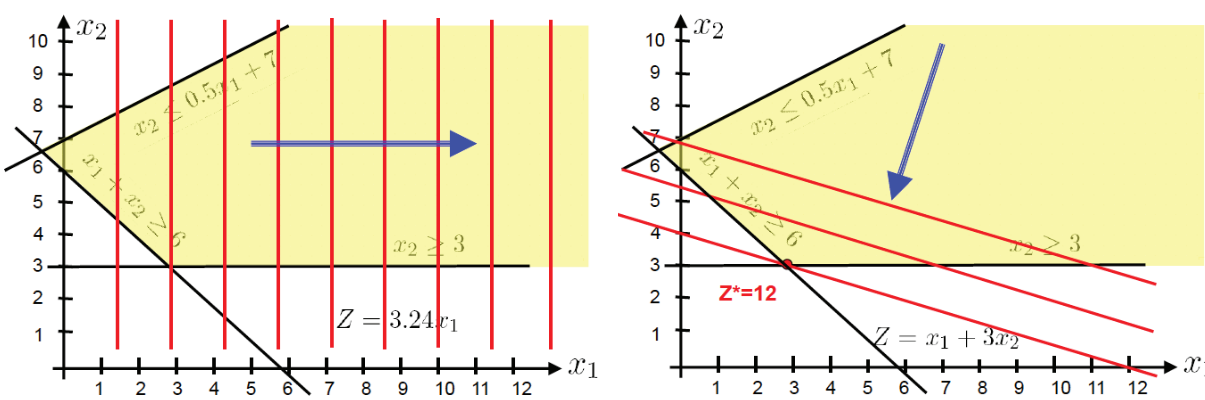


Figure 26: An unbounded [LEFT] and bounded [RIGHT] objective function.

This graphical analysis motivates the following proposition about LP solutions.

Proposition 5 (LP Solutions)**.** The solution to any linear program is characterized by one of the following three categories:

- **[No Solution]** This occurs when the feasible set is empty, or the objective function is unbounded.
- **[One Unique Solution]** There exists a single unique solution at the vertex of the feasible set. That is, at least two constraints are active and their intersection gives the optimal solution.
- **[A Non-Unique Solution]** There exists an infinite number of solutions, given by one edge of the feasible set. That is, one or more constraints are active and all solutions along the intersection of these constraints are equally optimal. This can only occur when the objective function gradient is orthogonal to one or multiple constraint.

Exercise 9. *Construct graphical examples of each of the three possible LP solutions given above.*

6 Appendix: Quadratic Programming (QP) in Detail

We begin our exposition of quadratic programming problems with the following example.

Example 6.1 (Linear Regression)**.** Suppose you have collected measured data pairs (x_i, y_i), for $i = 1, \cdots, N$ where $N > 6$, as shown in Fig. 27. You seek to fit a fifth-order polynomial to this data, i.e.

$$y = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4 + c_5 x^5 \tag{198}$$

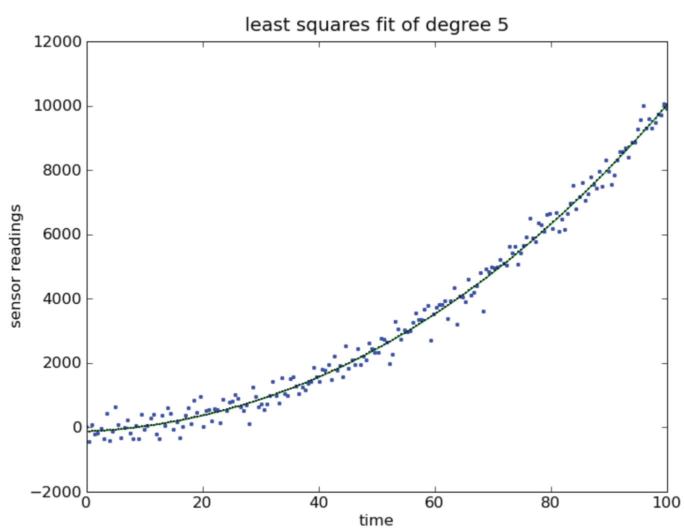


Figure 27: You seek to fit a fifth-order polynomial to the measured data above.

The goal is to determine parameters $c_i,\ i = 0, \cdots, 5$ that "best" fit the data in some sense. To this end, you may compute the residual r for each data pair:

$$c_0 + c_1 x_1 + c_2 x_1^2 + c_3 x_1^3 + c_4 x_1^4 + c_5 x_1^5 - y_1 \quad = \quad r_1,$$

$$\begin{aligned} c_0 + c_1 x_2 + c_2 x_2^2 + c_3 x_2^3 + c_4 x_2^4 + c_5 x_2^5 - y_2 &= r_2, \\ \vdots &= \vdots \\ c_0 + c_1 x_N + c_2 x_N^2 + c_3 x_N^3 + c_4 x_N^4 + c_5 x_N^5 - y_N &= r_N, \end{aligned} \tag{199}$$

which can be arranged into matrix-vector form $Ac - y = r$, where

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 & x_1^5 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & x_2^5 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 & x_N^3 & x_N^4 & x_N^5 \end{bmatrix}, \qquad c = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix}, \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \qquad r = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix}. \tag{200}$$

Now we compute an optimal fit for c in the following sense. We seek the value of c which minimizes the squared residual

$$\min_c \quad \frac{1}{2}\|r\|_2^2 = \frac{1}{2} r^T r = \frac{1}{2}(Ac - y)^T (Ac - y) = \frac{1}{2} c^T A^T A c - y^T A c + \frac{1}{2} y^T y. \tag{201}$$

Note that (201) is quadratic in variable c and therefore a convex function of c. This produces another special case of convex problems called **quadratic programs**. In this case the problem is unconstrained. As a result, we can set the gradient with respect to c to zero and directly solve for the minimizer.

$$\begin{aligned} \frac{\partial}{\partial c} \frac{1}{2}\|r\|_2^2 = A^T A c - A^T y = 0, \\ A^T A c = A^T y, \\ \boxed{c = (A^T A)^{-1} A^T y} \end{aligned} \tag{202}$$

This provides a direct formula for fitting the polynomial coefficients $c_i,\ i = 0, \cdots, 5$ using the measured data.

Exercise 10 (Tikhonov or L_2 regularization, a.k.a. Ridge Regression)**.** *Consider the fifth-order polynomial regression model in (198). Suppose we seek the value of c which minimizes the squared residual plus a so-called Tikhonov regularization term:*

$$\min_c \quad \frac{1}{2}\|r\|_2^2 + \frac{1}{2}\|\Gamma c\|_2^2. \tag{203}$$

for some matrix $\Gamma \in \mathbb{R}^{6\times 6}$. Derive the QP. Solve for the minimizer of this unconstrained QP. Provide a formula for the optimal coefficients c.

Exercise 11. *Consider fitting the coefficients c_1, c_2, c_3 of the following sum of radial basis functions to data pairs (x_i, y_i), $i = 1, \cdots, N$.*

$$y = c_1 e^{-(x-0.25)^2} + c_2 e^{-(x-0.5)^2} + c_3 e^{-(x-0.75)^2} \tag{204}$$

Formulate and solve the corresponding QP problem.

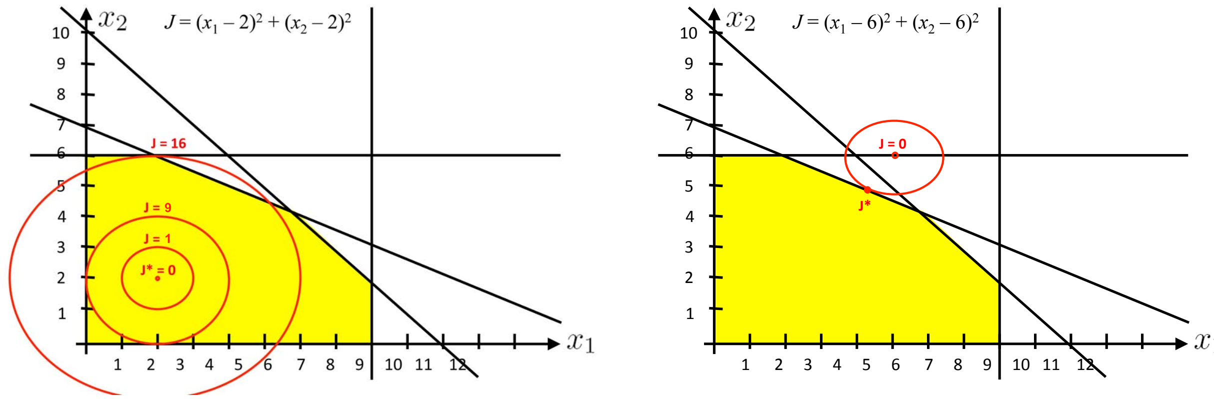


Figure 28: An interior optimum [LEFT] and boundary optimum [RIGHT] for a QP solved graphically.

Exercise 12. *Repeat the same exercise for the following Fourier Series:*

$$y = c_1 \sin(\omega x) + c_2 \cos(\omega x) + c_3 \sin(2\omega x) + c_4 \cos(2\omega x) \tag{205}$$

6.1 General Form of QP

Quadratic programs can be identified as having a quadratic objective function and linear constraints. Mathematically, this takes the form

$$\begin{aligned} \min \quad & \frac{1}{2} x^T Q x + R^T x, \\ \text{subject to:} \quad & Ax \leq b, \\ & A_{eq} x = b_{eq}. \end{aligned}$$

6.2 Graphical QP

For problems of one, two, or three dimensions, it is possible to solve QPs graphically. Consider the following QP example:

$$\begin{array}{llrcl} \min & & J = (x_1 - 2)^2 + (x_2 - 2)^2 \\ \\ \text{s. to} & & 2x_1 + 4x_2 & \leq & 28 \\ & & 5x_1 + 5x_2 & \leq & 50 \\ & & x_1 & \leq & 8 \\ & & x_2 & \leq & 6 \\ & & x_1 & \geq & 0 \\ & & x_2 & \geq & 0 \end{array}$$

The feasible set and corresponding iso-contours are illustrated in the left-hand side of Fig. 28. In this case, the solution is an **interior optimum.** That is, no constraints are active at the minimum. In contrast, consider the objective function $J = (x_1 - 6)^2 + (x_2 - 6)^2$ shown on the right-hand side of Fig. 28. In this case, the minimum occurs at the boundary and is unique.