Introduction to Numerical Optimization
Problem Sets with Solutions

Mathias Berger

June 2022

Contents

Acknowledgements

Most of these notes were written while serving as a teaching assistant for the *Introduction to Numerical Optimization* course taught to first and second-year graduate students by Pr. Quentin Louveaux in the Department of Electrical Engineering and Computer Science at the University of Liège, Belgium. They complement the teaching material covering the basic theory underpinning the classes of optimization models and methods discussed in this course. Appropriate reference to theoretical concepts and reminders are made in the worked-out solutions, which makes them reasonably self-contained and accessible to students and practitioners with a college-level grasp of calculus and linear algebra..

I would like to thank Quentin Louveaux for kindly accepting to review these notes and many fruitful exchanges. I would also like to thank Damien Gérard, who provided draft solutions to several of the problems treated in this document, and Adrien Bolland, who provided helpful comments on the first few problem sets. Several attentive students also helped improve the clarity and quality of these notes. All remaining errors are mine and can be reported at mathias.berger@alumni.duke.edu.

Part I

Problem Sets

Chapter 1

Introduction to Optimization Modelling

Problem 1

Your numerical optimization professor sometimes engages in business activities to earn a little extra cash. He runs a family business that produces and sells dairy products made from the milk of three family cows and he seeks to maximize his profits.

80 litres of milk are produced every week. The butter-making process requires 7 litres of milk to produce one kilogram of butter, whereas 3 litres of milk are necessary to produce one litre of ice cream.

Your professor is widely known in the milk business and always sells everything he produces. He sets the prices of his products to ensure revenues of 2€ per litre of ice cream and 7€ per kilogram of butter, and manages to keep all costs fixed.

Your professor owns a huge refrigerator that can store virtually unlimited amounts of butter, but his freezer can hold at most 20 litres of ice cream.

The family can work at most 6 hours every week to manufacture their delicious products. It takes one hour to produce 15 litres of ice cream and one hour to produce one kilogram of butter.

Problem 2

One wants to build an antenna relay to serve three villages. On a Cartesian map, the villages have coordinates $(0, 0), (1, 1), (2, 1)$. One wants to find the antenna location that minimizes the maximum distance from the antenna to these three villages. Extend this formulation to work with any number of villages.

Problem 3

We consider a grid-tied micro-grid comprising a number of devices and appliances. More precisely, the following systems are connected to the microgrid:

1. a number of appliances with total consumption C_t over each time period $t, t \in \mathcal{T} = \{1 \dots T\}$, of duration ΔT;
2. photovoltaic panels which deliver P_t^S watts;
3. a LiFePO$_4$-type battery to store electricity;
4. a hydrogen-based device to store energy, along with an electrolyzer transforming electricity into hydrogen, and a fuel cell converting hydrogen back into electricity.

The micro-grid is also connected to the distribution grid, such that an amount of electricity P_t^I can be purchased at any time period $t \in \mathcal{T}$. The LiFePO$_4$-type battery is a short-term storage device assumed to have a power density sufficient to accommodate the instantaneous power (no bound on the maximum flow) and a maximum energy storage capacity S^B. Unlike the battery, we consider the hydrogen storage system to have unlimited energy capacity. However, the input and output power flows are bound by F^{H_2}.

The micro-grid owner wishes to minimize the amount of electricity she must buy from the grid over the time horizon considered.

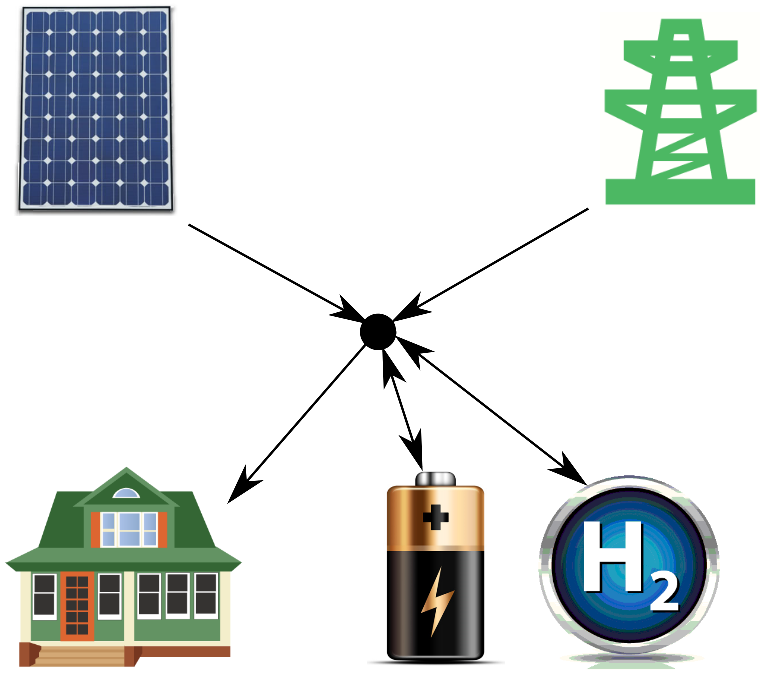


Figure 1.1: Schematic representation of the micro-grid.

Chapter 2

Linear Programming and the Primal Simplex Method

Problem 1

Consider the following linear optimization problem:

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

1. Sketch the problem and give its optimal solution.
2. Write the problem in standard form.
3. At the optimal solution, give the value of each variable of the problem in its standard form.
4. Solve the problem with the simplex method.

Problem 2

Consider the polytope

$$\begin{array}{rcrcl} 2x_1 & + & 3x_2 & = & 7 \\ x_1 & - & x_2 & \leq & 0 \\ 3x_1 & + & 2x_2 & \geq & 2 \\ x_1 & , & x_2 & \geq & 0. \end{array}$$

Plot the feasible set, describe each primal feasible basis, along with the corresponding basic feasible solutions.

Problem 3

Consider the following optimization problem:

$$\begin{array}{lrcrcrcrcrcl} \max & 4x_1 & + & 7x_2 & + & 3x_3 & + & x_4 & + & 5x_5 & & \\ \text{s.t.} & 2x_1 & + & 3x_2 & + & x_3 & & & + & 3x_5 & \leq & 19 \\ & x_1 & - & x_2 & & & + & 2x_4 & & & \geq & 0 \\ & x_1 & + & 3x_2 & & & + & x_4 & + & 2x_5 & \leq & 10 \\ & x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & \geq & 0. \end{array}$$

The optimal solution of the problem is $\mathbf{x}^\star = (x_1^\star, x_2^\star, x_3^\star, x_4^\star, x_5^\star) = (0, 0, 19, 10, 0)$. Build the optimal simplex tableau.

Problem 4

Consider the following linear program in standard form:

$$
\begin{array}{lcccccccccccc}
\min & & - & x_2 & & & & & & & & & \\
\text{s.t.} & x_1 & + & x_2 & - & x_3 & & & & & = & 1 \\
 & 2x_1 & - & x_2 & & & - & x_4 & & & = & 2 \\
 & 2x_1 & + & x_2 & & & & & + & x_5 & = & 6 \\
 & x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & \geq & 0
\end{array}
$$

1. State the *Big M* optimization problem.
2. Perform the *phase I* procedure for finding an initial primal feasible solution. State the *phase II* problem.

Chapter 3

The Revised Simplex, LP Duality and the Dual Simplex

Problem 1

Consider the following linear program:

$$\begin{array}{rrcrcrcrcrcl}
\min & x_1 & - & x_2 & + & x_3 & & & & & & \\
\text{s.t.} & x_1 & & & - & x_3 & & & & & \leq & 1, \\
& 2x_1 & + & x_2 & & & & & + & x_5 & \leq & 2, \\
& x_1 & + & x_2 & - & x_3 & - & x_4 & & & \leq & 3, \\
& & & & & & & x_4 & + & x_5 & \leq & 4, \\
& & & & & x_3 & + & 2x_4 & + & 2x_5 & \leq & 5, \\
& x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & \geq & 0.
\end{array}$$

We will apply one iteration of the revised simplex algorithm. For the sake of illustration, we will use the inverse basis matrix, keeping in mind that the LU factorization of the basis matrix is used in modern implementations instead. We denote by s_1, s_2, s_3, s_4, s_5 the slack variables of all five constraints.
We consider that x_1, x_2, s_3, s_4, s_5 are the current basic variables. The corresponding inverse basis matrix is given by

$$A_B^{-1} = \begin{pmatrix} 1 & & & & \\ -2 & 1 & & & \\ 1 & -1 & 1 & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix}$$

1. Compute the value of the basic variables at the vertex represented by the current basis.
2. Compute the reduced costs of all nonbasic variables. Which variable can enter the basis?
3. Consider the variable that enters the basis. Compute its column in the simplex tableau. Which variable should leave the basis?
4. Compute the new inverse matrix and the new value of the vertex after a pivot.

Problem 2

Write the dual of the following problem

$$\begin{array}{rl}
\min & x_1 - x_2 \\
\text{s.t.} & 2x_1 + 3x_2 - x_3 + x_4 \leq 0 \\
& 3x_1 + x_2 + 4x_3 - 2x_4 \geq 3
\end{array}$$

$$\begin{aligned} &-x_1 - x_2 + 2x_3 + x_4 = 6 \\ &x_1 \leq 0 \\ &x_2, x_3 \geq 0 \end{aligned}$$

Problem 3

Consider the following linear program:

$$\begin{array}{cl} \min & 2x_1 + x_2 \\ \text{s.t.} & -2x_1 + x_2 \leq -1 \\ & x_1 - 3x_2 \leq -2 \\ & x_1 + x_2 \leq 5 \end{array}$$

Prove, without solving the problem, that the solution $x^* = (1, 1)$ is optimal.

Problem 4

Solve the problem

$$\begin{array}{cl} \min & 2x_1 + x_2 \\ \text{s.t.} & -x_1 - 2x_2 \leq -3 \\ & -x_1 - x_2 \leq -2 \\ & -x_1 + x_2 \leq 1 \end{array}$$

with the dual simplex.

Chapter 4

Sensitivity Analysis in Linear Programming

Problem 1

Given the linear problem

$$\begin{array}{lccccccc}
\max & 2x_1 & + & 5x_2 & + & x_3 & & \\
\text{s.t.} & x_1 & & & + & x_3 & \geq & 1 \\
 & & - & x_2 & + & 2x_3 & \geq & 0 \\
 & 3x_1 & & & + & x_3 & \leq & 5 \\
 & x_1 & , & x_2 & , & x_3 & \geq & 0
\end{array}$$

whose optimal tableau is

$$\begin{array}{cccccc|c}
x_1 & x_2 & x_3 & s_1 & s_2 & s_3 & b \\
\hline
31 & 0 & 0 & 0 & 5 & 11 & \\
\hline
3 & 0 & 1 & 0 & 0 & 1 & 5 \\
6 & 1 & 0 & 0 & 1 & 2 & 10 \\
2 & 0 & 0 & 1 & 0 & 1 & 4
\end{array}$$

1. How does the optimal solution evolve if we replace the right-hand side coefficient of the second inequality by 2? Does the basis remain unchanged?

2. Determine for which x_3 cost values the optimal basis remains unchanged.

3. The constraint $x_1 - x_2 \geq 5$ is added to the initial problem. Compute the new optimal solution.

Problem 2

A PCB foundry can produce 4 types of processors with different architectures, namely Arrandale (i3), Clarkdale (i5), Penryn (Core 2M) and Bloomfield (i7). To manufacture these processors, silicon wafers are subject to photolithography, etching and doping processes. Two different procedures can be employed to produce Penryn processors, but technical constraints imply that they must be run in parallel.

Resource requirements and available quantities are shown below, along with the expected revenue from the sale of each type of processor.

Ressource	A	C	P_1	P_2	B	Total Max.
Silicon (kg)	10	15	10	10	20	130
Photolithography (hours)	1	2	2	1	1	13
Etching (hours)	3	1	6	6	3	45
Doping (hours)	2	4	2	5	3	23
Revenue	51	102	66	66	89	/

Then, the revenue maximization problem of the manufacturer is given below. Decision variables represent the quantities of processors of Arrandale (A), Clarkdale (C), Penryn method 1 (P1), Penryn method 2 (P2) and Bloomfield (B) types that should be manufactured, respectively. Though in a practical context, we would expect these variables to take integer values, for the sake of simplicity, real variables are considered in this problem.

$$\begin{array}{lrcrcrcrcrcr}
\max & 51A & + & 102C & + & 66P_1 & + & 66\,P_2 & + & 89B & & \\
\text{s.t.} & 10A & + & 15C & + & 10P_1 & + & 10P_2 & + & 20B & \leq & 130 \\
 & A & + & 2C & + & 2P_1 & + & P_2 & + & B & \leq & 13 \\
 & 3A & + & C & + & 6P_1 & + & 6P_2 & + & 3B & \leq & 45 \\
 & 2A & + & 4C & + & 2P_1 & + & 5P_2 & + & 3B & \leq & 23 \\
 & & & & & P_1 & - & P_2 & & & = & 0 \\
\end{array}$$
$$A,\ C,\ P_1,\ P_2,\ B,\ \geq 0.$$

The solution of the primal and dual problems, respectively, and the sensitivity analysis report are shown below:

	Optimal value	Reduced cost	Objective coefficient	Allowed increase	Allowed decrease
A	0	-3.571	51	3.571	∞
C	2	0	102	16.667	12.5
P_1	0	0	66	37.571	∞
P_2	0	-37.571	66	37.571	∞
B	5	0	89	47	12.5

Table 1. Optimal solution of the primal problem and its sensitivity to a change in objective function coefficients. The last two columns give the changes in the objective function coefficients allowing to keep the optimal basis unchanged.

	Slack variable	Dual variable	b	Allowed increase	Allowed decrease
Silicon	0	1.429	130	23.33	43.75
Photol.	4	0	13	∞	4
Etching	28	0	45	∞	28
Doping	0	20.143	23	5.60	3.50
$P_1 = P_2$	0	11.429	0	3.50	0

Table 2. Optimal solution of the dual problem and its sensitivity. The second column gives the optimal value of the slack variable of each primal constraint. The last two columns give the changes in primal right-hand side coefficients allowing to keep the same solution to the dual problem.

Answer the following questions using these tables:

1. What is the optimal quantity of each processor type and what is the total revenue?
2. Give an economic interpretation of the optimal dual variables.
3. Should the manufacturer buy an additional 20 kilograms of silicon at 1.1 \$/kg?
4. Suppose that the number of available hours in the etching room decreases by 30. What can be said about the decrease in revenue?

BONUS: In this model, the quantities of Penryn method 1 and Penryn method 2 must be equal. Consider an updated model in which this constraint is replaced by the constraint $P_1 - P_2 \geq 0$. In this reformulated problem, is the quantity of produced Penryn method 1 positive?

Chapter 5

Convex Models

Problem 1

a. Let $g : \mathbb{R}^n \mapsto \mathbb{R}_+$ be a norm. Show that g is a convex function.

b. Show that the intersection of two convex sets is a convex set.

c. Show that the intersection of two cones is a cone.

d. Let $g : \mathbb{R} \mapsto \mathbb{R}$ be a strictly increasing function, that is, $g(y) < g(z)$ if $y < z$, and let $h_1 : \mathbb{R}^n \mapsto \mathbb{R}$ and $h_2 : \mathbb{R}^n \mapsto \mathbb{R}$ be arbitrary functions. In addition, let $S_h = \{x \in \mathbb{R}^n | \ h_1(x) \leq h_2(x)\}$ and $S_g = \{x \in \mathbb{R}^n | \ g(h_1(x)) \leq g(h_2(x))\}$. Show that $S_h = S_g$.

e. Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ and let $\text{epi}(f) = \{(x, t) \in \mathbb{R}^{n+1} | \ f(x) \leq t\}$ be the *epigraph* of f. In addition, let $F \subset \mathbb{R}^n$. Show that $\min\{t \mid (x, t) \in \text{epi}(f), \ x \in F\}$ and $\min\{f(x) \mid x \in F\}$ are equivalent.

f. Let $g : \mathbb{R} \mapsto \mathbb{R}$ be a strictly increasing function, let $h : \mathbb{R}^n \mapsto \mathbb{R}$ be an arbitrary function, and let $F \subset \mathbb{R}^n$. In addition, let $S_h = \arg\max\{h(x) | \ x \in F\}$ and let $S_g = \arg\max\{g(h(x)) | \ x \in F\}$. Show that $S_h = S_g$.

g. Let $f : \mathbb{R} \mapsto \mathbb{R}$ be a strictly decreasing function, i.e., $f(x) < f(y)$ if $x > y$, and let $h : \mathbb{R} \mapsto \mathbb{R}$ be an arbitrary function. In addition, let $F = \{x \in \mathbb{R}^n_+ | \ Ax \leq b\}$, let $S_f = \arg\max\{f(h(x)) | \ x \in F\}$ and let $S_h = \arg\min\{h(x) | \ x \in F\}$. Show that $S_f = S_h$.

Problem 2

Show that the following optimisation problems are convex (or not). Where possible, propose a convex reformulation with the same optimal solution set.

$$\text{(a)} \quad \begin{aligned} \min \ & |x_1 - 3x_2| \\ \text{s.t.} \ & x_1 + 2x_2 \leq 3 \\ & -2x_1 + 3x_2 \leq 0 \\ & x_1 \geq 2, x_2 \geq 0 \end{aligned}$$

$$\text{(b)} \quad \begin{aligned} \max \ & |x_1 + 2x_2| \\ \text{s.t.} \ & Ax = b \\ & x \in \mathbb{R}^n_+ \end{aligned}$$

$$\text{(c)} \quad \begin{aligned} \max \ & |x_1 - 2x_2| \\ \text{s.t.} \ & Ax = b \\ & x \in \mathbb{R}^n_+ \end{aligned}$$

(d)
$$\begin{aligned} \min\ & \sqrt{x_1^2 + x_2^2} \\ \text{s.t.}\ & x_1 + x_2 = \frac{1}{2} \end{aligned}$$

(e)
$$\begin{aligned} \min\ & x_1 + 2x_2 \\ \text{s.t.}\ & x_1^2 + 9x_2^2 + x_3^2 \le 4 \end{aligned}$$

(f)
$$\begin{aligned} \min\ & 2x_1 - x_2 \\ \text{s.t.}\ & (x_1 - 3x_2)^3 \le (2x_1 + x_2)^3 \\ & x \in \mathbb{R}^2 \end{aligned}$$

(g)
$$\begin{aligned} \max\ & x_1 x_2 \\ \text{s.t.}\ & Ax = b \\ & x \in \mathbb{R}^n_+ \end{aligned}$$

(h)
$$\begin{aligned} \min\ & 1 \\ \text{s.t.}\ & \frac{||Ax + b||^2}{c^T x + d} \le t \\ & c^T x + d > 0 \end{aligned}$$

(i)
$$\begin{aligned} \max\ & \frac{1}{c^T x} \\ \text{s.t.}\ & Ax \le b \\ & x \in \mathbb{R}^n \end{aligned}$$

(j)
$$\begin{aligned} \min\ & \log(c^T x) \\ \text{s.t.}\ & Ax \le b \\ & x \in \mathbb{R}^n_+ \end{aligned}$$

(k)
$$\begin{aligned} \min\ & \lambda_{\max}(X) \\ & AX = B \\ & X = X^T \\ & X \in \mathbb{R}^{n \times n} \end{aligned}$$

(l)
$$\begin{aligned} \min\ & \lambda_{\max}(X) + x_{11}^2 \\ & AX = B \\ & X = X^T \\ & X \in \mathbb{R}^{n \times n} \end{aligned}$$

(m)
$$\begin{aligned} \min\ & \max\{\lambda_{\max}(X), x_{11}^2\} \\ & AX = B \\ & X = X^T \\ & X \in \mathbb{R}^{n \times n} \end{aligned}$$

Problem 3

Let $Q \in \mathbb{R}^{n \times n}$ be a positive definite matrix, i.e., $Q = Q^T$ and $\lambda_i(Q) > 0,\ i = 1, \ldots, n$. Transform the following optimisation problem into a conic program

$$\begin{array}{rl} \min & x^T Q x \\ \text{s.t.} & ||x||_2 \leq \gamma \end{array}$$

Chapter 6

Second-Order Cone Programming

Problem 1

One wants to build an antenna relay to serve three villages. The Cartesian coordinates of the villages are $(0,0), (1,1)$ and $(2,1)$, respectively. One wants to find the location that minimises the maximum distance from the antenna to these three villages. Formulate this problem as a conic program.

Problem 2

The goal of James Bond's latest mission is to defuse a bomb planted on a boat by his worst enemy. Bond (A) is on the beach, 50 metres away from the shore, while the boat (B) is in the water, 50 metres away from the shore. In addition, the boat is 100 metres to the right of Bond, as shown in Figure 6.1. Bond, who runs at 5 m/s and swims at 2.778 m/s, is trying to figure out the fastest path to the boat.

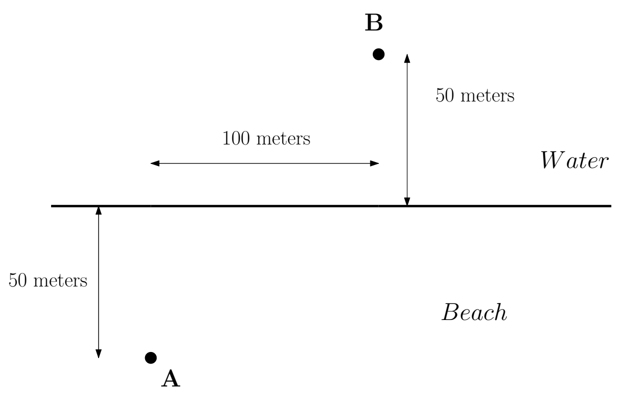


Figure 6.1: Schematic of Bond's mission.

a. Formulate the problem as a conic programming problem.

b. Write the dual formulation.

Problem 3

Let $c \in \mathbb{R}^n$ be a cost vector and $b_i \in \mathbb{R},\ i = 1, \ldots, N$, be a set of (deterministic) scalar parameters. Let $a_i \in \mathbb{R}^n,\ i = 1, \ldots, N$, be independent Gaussian random vectors with mean $\bar{a}_i \in \mathbb{R}^n,\ i = 1, \ldots, N$, and covariance matrix $\Sigma_i \in \mathbb{R}^{n \times n},\ i = 1, \ldots, N$. In addition, let $\mathbb{P}$ be a probability measure and let

$\eta \in [0.5, 1]$ denote a confidence level. Show that the linear program with probabilistic constraints,

$$\begin{aligned} \min\ & c^T x \\ \text{s.t. } & \mathbb{P}(a_i^T x \le b_i) \ge \eta,\ i = 1, \dots, N \\ & x \in \mathbb{R}^n, \end{aligned}$$

which is also known as a *chance constrained linear program*, can be cast as a second-order cone program.

Problem 4

Let $X = \{x_1, \dots, x_N\}$ and $Y = \{y_1, \dots, y_M\}$ be two sets of points in $\mathbb{R}^n$. We seek an affine function $f : \mathbb{R}^n \mapsto \mathbb{R}$, $f(z) = w^T z - b$, that linearly separates X and Y, that is, such that $f(x_i) \ge 1,\ i = 1, \dots, N$, and $f(y_j) \le -1,\ j = 1, \dots, M$. Put differently, we seek $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ such that X and Y are separated by the two hyperplanes $w^T z - b = 1$ and $w^T z - b = -1$. Since perfect linear separation is not always possible, we allow these constraints to be violated, but we would like the amount by which they are violated to be as small as possible. Moreover, to produce a robust classifier, we want the distance between hyperplanes to be as large as possible. In other words, we seek to balance classifier accuracy and robustness.

a. For each point in X and Y, express the penalty resulting from a classification error in terms of the max function, and suggest a criterion that will increase classifier robustness.

b. Formulate the resulting optimisation problem.

c. Show that the problem at hand can be cast as a second-order cone program.

Chapter 7

Semidefinite Programming

Problem 1

Solve the problem

$$\begin{array}{ll} \min & 2p_1 + p_4 \\ \text{s.t.} & \begin{pmatrix} p_1 & p_2 & p_3 \\ p_2 & p_4 - 1 & 0 \\ p_3 & 0 & p_4 \end{pmatrix} \succcurlyeq 0. \end{array}$$

Problem 2

Let $p : \mathbb{R} \mapsto \mathbb{R}$ be the polynomial

$$p(x) = 2x^4 - 2x^3 + 4x^2 - 2x + 2.$$

1. Formulate a semidefinite program to check whether p is a nonnegative polynomial.
2. Write the dual problem of the SDP found above.
3. Formulate a second-order cone program to check whether p is a nonnegative polynomial.
4. Formulate a linear program to check whether p is a nonnegative polynomial.

Chapter 8

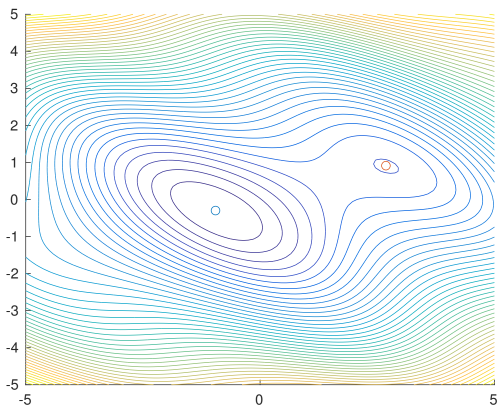
Unconstrained Optimization and Descent Methods

Problem 1

Let $g : \mathbb{R}^2 \mapsto \mathbb{R}$ be a function such that $g(x, y) = \sin(x + y) + \frac{x^2}{6} + \frac{y^2}{2}$, which has local minima $(-0.9393, -0.3131)$ and $(2.6965, 0.8988)$. Let $(x_0, y_0) = (0, 0)$ and compute

a. the gradient descent direction at (x_0, y_0).

b. the Newton direction at (x_0, y_0).



Problem 2

Let $g : \mathbb{R}^2 \mapsto \mathbb{R}$ be a continuously differentiable function such that

$$g(z) = g(z_1, z_2) = \frac{(z_1 + z_2 - 2)^2}{2} + (z_1 - z_2 + 2)^2,$$

and consider the unconstrained minimisation problem

$$\min_{z \in \mathbb{R}^2} \quad g(z).$$

Sketch an alternating minimisation scheme which does not require the computation of the full gradient at each iteration.

Problem 3

Consider the unconstrained minimisation problem

$$\min_{x,y \in \mathbb{R}} \quad (y - x^2)^2 + (1 - x)^2,$$

and perform one step of the gradient descent algorithm using the Wolfe conditions to identify the step size.

Chapter 9

Constrained Optimization and Interior Point Methods

Problem 1

1. Determine the central path of the following optimization problem :

$$\begin{array}{rl} \min & x_1 - 2x_2 \\ s.t. & x_1 + 2x_2 = 1 \\ & x_1, x_2 \geq 0. \end{array}$$

2. What is the optimal solution ?
3. What is the analytical center of the polyhedron ?
4. Write the dual of the problem.
5. Show that the primal-dual central path satisfies the modified KKT conditions.

Chapter 10

Automatic Differentiation

Problem 1

Let $f : \mathbb{R}^2 \mapsto \mathbb{R}$ be the mapping $f(x, y) = x^2 + \frac{\sin(xy)}{x}$. Compute the gradient of f using

(a) the forward accumulation automatic differentiation algorithm ;

(b) the reverse accumulation automatic differentiation algorithm.

Part II

Solutions

Chapter 11

Introduction to Optimization Modelling

Solution to Problem 1

We need two optimization variables representing the amount of butter and ice cream produced, respectively. Let $x_1 \in \mathbb{R}_+$ denote the amount of butter produced and let $x_2 \in \mathbb{R}_+$ be the amount of ice cream produced. From the problem statement, we deduce that three constraints should be considered. The first constraint pertains to the amount of milk available to produce butter and ice cream every week. Obviously, the family cannot use more milk than what it has at its disposal, hence

$$7x_1 + 3x_2 \leq 80.$$

The second constraint expresses the fact that the amount of ice cream that may be produced is limited by the freezer capacity,

$$x_2 \leq 20.$$

The third constraint enforces that the family cannot spend more than 6 hours producing butter and ice cream every week,

$$x_1 + \frac{1}{15}x_2 \leq 6.$$

The goal of the family is to maximise its revenue. Since costs remain fixed, they do not have any impact on the amount of butter and ice cream that should be produced (although they will influence the profits made by the family). The revenue of the family can thus written as

$$7x_1 + 2x_2,$$

such that the full problem reads

$$\begin{array}{rrcrcr} \max\limits_{x_1,x_2} & 7x_1 & + & 2x_2 & & \\ \text{s.t.} & 7x_1 & + & 3x_2 & \leq & 80 \\ & & & x_2 & \leq & 20 \\ & x_1 & + & \frac{1}{15}x_2 & \leq & 6 \\ & x_1 & , & x_2 & \geq & 0 \end{array}$$

As will be seen later, this particular problem is an instance of a linear program.

Solution to Problem 2

We work in the general setting directly. Let us assume that there are n villages with coordinates $\bar{x}^j = \begin{pmatrix}\bar{x}_1^j & \bar{x}_2^j\end{pmatrix}, j = 1, \ldots, n$, and let $x \in \mathbb{R}^2$ denote the coordinates of the antenna. The Euclidean distance $d_j : \mathbb{R}^2 \to \mathbb{R}_+$ between the antenna and village j can be written as

$$d_j(x) = \sqrt{\sum_{i=1}^{2} \left(x_i - \bar{x}_i^j\right)^2}.$$

We then seek to identify the location of the antenna x so as to minimise the maximum distance between the antenna and any village, that is,

$$\min_x \max\{d_j(x) | j = 1, \dots, n\}.$$

Note that minimising the maximum distance is equivalent to minimising an auxiliary (scalar) variable that bounds all distances from above. The problem can thus be re-formulated as

$$\begin{aligned} \min_{x,d} \ & d \\ \text{s.t.} \ & d_j(x) \leq d, j = 1, \dots, n, \\ & x \in \mathbb{R}^2, d \in \mathbb{R}. \end{aligned}$$

As will be seen later in the course, this problem is an instance of a second-order cone program.

Solution to Problem 3

Let $P^I_t \in \mathbb{R}_+$ denote the amount electricity imported from the grid at time t. Let $P^S_t \in \mathbb{R}_+$ be the electricity producer by solar panels at time t. The maximum amount of electricity that solar panels may produce at time t is given by κ^S_t. Curtailment is assumed to be allowed so that κ^S_t bounds P^S_t from above,

$$P^S_t \leq \kappa^S_t, \forall t \in \mathcal{T}.$$

Let $E^B_t \in \mathbb{R}_+$ be the energy stored in the battery at time t and let $P^B_t \in \mathbb{R}$ be the power charged/discharged from the battery at time t. The state-of-charge dynamics of the battery can be expressed as

$$E^B_{t+1} = E^B_t - P^B_t, \forall t \in \mathcal{T} \setminus \{|\mathcal{T}|\},$$

where by convention $P^B_t > 0$ indicates that the battery is being discharged. The amount of energy stored in the battery is bounded from above by S_B, thus

$$E^B_t \leq S^B, \forall t \in \mathcal{T}.$$

Now, let $E^{H_2}_t \in \mathbb{R}_+$ be the energy stored in the hydrogen storage system at time t and let $P^{H_2}_t \in \mathbb{R}$ be the power charged/discharged from it at time t. Assuming that the storage and conversion is lossless, the storage dynamics can be expressed in a fashion similar to that of the battery storage system,

$$E^{H_2}_{t+1} = E^{H_2}_t - P^{H_2}_t, \forall t \in \mathcal{T} \setminus \{|\mathcal{T}|\}.$$

The amount of energy that can be withdrawn from the hydrogen storage system at time t is limited by F^{H_2}, such that

$$-F^{H_2} \leq P^{H_2}_t \leq F^{H_2}, \forall t \in \mathcal{T}.$$

Finally, the electricity available in the microgrid at any point in time should be equal to the electricity consumption of all appliances and devices,

$$P^S_t + P^I_t + P^B_t + P^{H_2}_t = C_t, \forall t \in \mathcal{T}.$$

Since the owner of the microgrid seeks to minimise the amount of electricity imported from the grid, the full problem reads

$$\begin{aligned} \min \ & \sum_{t \in \mathcal{T}} P^I_t \\ \text{s.t.} \ & P^S_t + P^I_t + P^B_t + P^{H_2}_t = C_t, \forall t \in \mathcal{T}, \\ & P^S_t \leq \kappa^S_t, \forall t \in \mathcal{T}, \\ & E^B_{t+1} = E^B_t - P^B_t, \forall t \in \mathcal{T} \setminus \{|\mathcal{T}|\}, \\ & E^B_t \leq S^B, \forall t \in \mathcal{T}, \\ & E^{H_2}_{t+1} = E^{H_2}_t - P^{H_2}_t, \forall t \in \mathcal{T} \setminus \{|\mathcal{T}|\}, \\ & -F^{H_2} \leq P^{H_2}_t \leq F^{H_2}, \forall t \in \mathcal{T}, \\ & P^S_t \in \mathbb{R}_+, E^B_t \in \mathbb{R}_+, E^{H_2}_t \in \mathbb{R}_+, \\ & P^I_t \in \mathbb{R}_+, P^B_t \in \mathbb{R}, P^{H_2}_t \in \mathbb{R}. \end{aligned}$$

Chapter 12

Linear Programming and the Primal Simplex Method

Solution to Problem 1

1. A schematic of the problem at hand is shown in Figure 12.1. In Figure 12.1, the black lines represent the sets of points satisfying inequality constraints in (2.1) with equality, i.e., $\mathbf{a}_i^T\mathbf{x} = b_i,\ i = 1,2,3$, with, e.g., $\mathbf{a}_1 = (-1,1)^T$ and $b_1 = 2$. The lines corresponding to nonnegativity constraints coincide with the horizontal and vertical axes. In addition, the blue arrows, which are proportional to the $\mathbf{a}_i$'s, represent vectors normal to the aforementioned hyperplanes and point in the direction of the feasible half-space defined by each inequality constraint. Thus, the feasible region of this optimization problem corresponds to the intersection of all feasible half-spaces and is shown in grey. Finally, the red arrow indicates the direction in which the objective function increases. Hence, it is clear that the optimal solution is unique and corresponds to the C vertex, which has coordinates $C = (x_1^\star, x_2^\star) = (2,4)$.

2. A linear optimization problem is said to be in standard form if it satisfies three conditions, namely,

1. It is a minimization problem.
2. All constraints except variable ranges are equality constraints.
3. All variables are nonnegative.

Clearly, problem (2.1) is not in standard form. To be expressed in standard form, it must undergo a series of transformations. First, it can be turned into a minimization problem by observing that

$$\max x_2 = -\min -x_2.$$

Then, each constraint $\mathbf{a}_i^T\mathbf{x} \le b_i,\ i = 1,2,3$, can be turned into an equality constraint by introducing a nonnegative slack variable $s_i \in \mathbb{R}_+$, such that $\mathbf{a}_i^T\mathbf{x} + s_i = b_i,\ i = 1,2,3$. A slack variable essentially expresses the difference between the terms on the right and left-hand sides of the corresponding constraint, and will take a nonzero value if this constraint is not tight. Combining these observations allows us to write problem (2.1) in standard form as

$$\begin{array}{rrcrcrcrcrcl} -\min & & & -x_2 & & & & & & & & \\ \text{s.t.} & -x_1 & + & x_2 & + & s_1 & & & & & = & 2 \\ & x_1 & + & x_2 & & & + & s_2 & & & = & 6 \\ & x_1 & & & & & & & + & s_3 & = & 4 \\ & x_1 & , & x_2 & , & s_1 & , & s_2 & , & s_3 & \ge & 0 \end{array} \tag{12.1}$$

3. Since $(x_1^\star, x_2^\star) = (2,4)$, it is straightforward to obtain the optimal values of the slack variables. Indeed, considering the first constraint directly yields

$$-2 + 4 + s_1^\star = 2 \Rightarrow s_1^\star = 0.$$

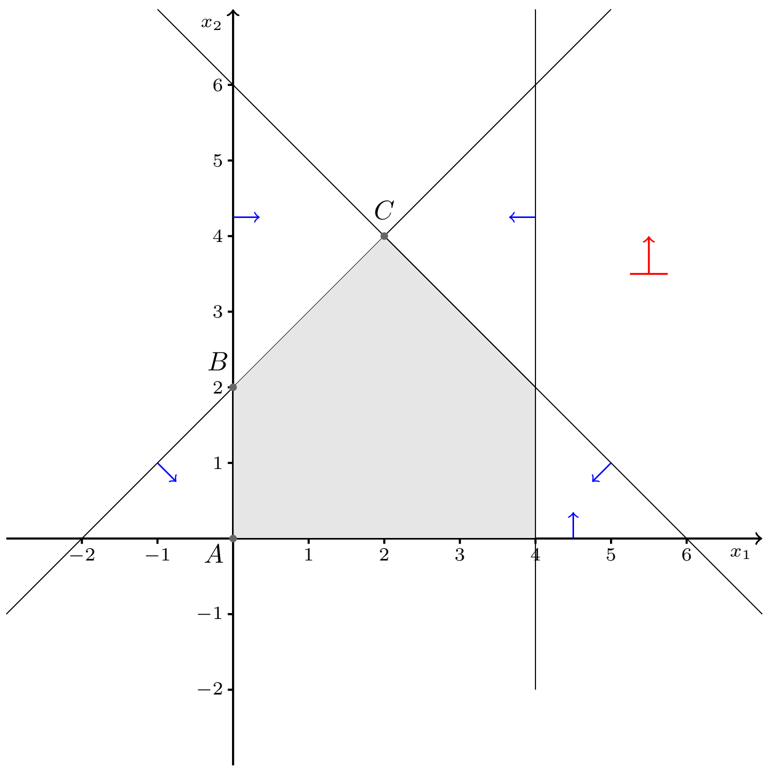


Figure 12.1: Schematic of problem (2.1). The feasible region is shown in grey, while the red arrow indicates the direction of increase of the objective function. The C vertex is optimal.

Likewise, for the other constraints,

$$2 + 4 + s_2^\star = 6 \Rightarrow s_2^\star = 0,$$

and

$$2 + s_3^\star = 4 \Rightarrow s_3^\star = 2,$$

respectively. Hence, the optimal solution of the problem in standard form is $(x_1^\star, x_2^\star, s_1^\star, s_2^\star, s_3^\star) = (2, 4, 0, 0, 2)$. At this stage, two comments are in order. Firstly, it is worth noticing that the first two constraints are tight at optimality. As a result, the corresponding slack variables are equal to 0, which is consistent with the claim made when introducing them (see **2.**). Secondly, the number of variables taking a nonzero value at optimality is equal to the number of equality constraints. Unless the problem is degenerate, this will always be the case. This property of solutions will become more salient in **4.**, where the primal simplex method is employed to solve (12.1).

4. Roughly speaking, the primal simplex algorithm is an iterative method visiting a sequence of vertices in order to identify an optimal solution or show that the problem is unbounded. The method relies crucially on the concept of a *basis*, which characterises vertices (though two bases may correspond to the same vertex), and a so-called *simplex tableau*, which offers a handy representation of bases and their properties. Starting from an initial tableau, the latter is progressively updated until optimality conditions are satisfied, as described in the following. In this case, the initial tableau reads as

x_1	x_2	s_1	s_2	s_3	
0	-1	0	0	0	
-1	1	1	0	0	2
1	1	0	1	0	6
1	0	0	0	1	4

The very first row of the tableau lists all variables, whereas the second row contains the (reduced) costs. Each of the other rows corresponds to a constraint, and the columns forming a permutation matrix, that is, a matrix obtained by permuting rows or columns of the identity matrix, correspond to variables forming the current basis, provided that their reduced costs are also equal to 0. Hence, in this case, the basis is formed by $\{s_1, s_2, s_3\}$, which are referred to as *basic variables*, while the remaining variables are called *nonbasic variables.* The very last column of the tableau shows the values of basic variables. This basis is said to be *primal feasible*, as all basic variables have nonnegative values, and therefore defines a *basic feasible solution*, which is equivalent to a vertex. By contrast, the nonbasic variables, namely $\{x_1, x_2\}$, all take on zero values. From a geometric perspective, this solution corresponds to point A in Figure 12.1. Then, from this tableau, the method seeks to update the basis, i.e., replace a basic variable by a nonbasic variable, so as to improve the objective function value. This operation is sometimes referred to as a *pivot.* In essence, selecting a new basic variable consists in identifying a stepping direction that yields an updated solution satisfying all equality constraints and improving the value of the objective function. Such a direction is referred to as a *basic direction.* In particular, if no basic direction can be found and the basis is primal feasible, then it is also optimal and the problem is solved. However, if such a direction is found and the step size can be set to an arbitrarily large value while keeping all variables nonnegative, this direction corresponds to an *extreme ray* of the polyhedron represented by the constraints, and the problem is unbounded. By contrast, if a basic direction is found and progressively increasing the step size causes some variables to become negative, a pivot must be performed. More precisely, the nonbasic variable corresponding to this direction should enter the basis and the first basic variable whose nonnegativity constraint becomes tight when stepping in this direction should exit the basis, defining the step size in the process. It is worth noting that, by construction, this procedure will always produce basic feasible solutions. More formally, let

$$A_{B1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } A_{N1} = \begin{pmatrix} \mathbf{A}_{x_1} & \mathbf{A}_{x_2} \end{pmatrix} = \begin{pmatrix} -1 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix},$$

be the initial basic and nonbasic matrices, respectively, such that the coefficient matrix can be expressed as $A = \begin{pmatrix} A_{N1} & A_{B1} \end{pmatrix}$. In this case, since the basic matrix is the identity matrix, A_{B1} and A_{N1} also store the coefficients of basic and nonbasic variables in the simplex tableau, respectively. Let

$$\mathbf{c}_{B1} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T \text{ and } \mathbf{c}_{N1} = \begin{pmatrix} c_{x_1} & c_{x_2} \end{pmatrix}^T = \begin{pmatrix} 0 & -1 \end{pmatrix}^T$$

be the objective coefficients of basic and nonbasic variables, respectively. Finally, let

$$\mathbf{b}_1 = \begin{pmatrix} 2 & 6 & 4 \end{pmatrix}^T \text{ and } \mathbf{x}_1 = \begin{pmatrix} 0 & 0 & 2 & 6 & 4 \end{pmatrix}^T$$

denote the vector of right-hand side coefficients and the current solution, respectively. Since there are two nonbasic variables, two basic directions are possible, namely

$$\mathbf{d}_{x_1} = \begin{pmatrix} 1 & 0 & \mathbf{d}_{B,x_1}^T \end{pmatrix}^T \text{ and } \mathbf{d}_{x_2} = \begin{pmatrix} 0 & 1 & \mathbf{d}_{B,x_2}^T \end{pmatrix}^T.$$

As suggested earlier, for any step size $\theta > 0$, the updated solution must satisfy all equality constraints, hence $A(\mathbf{x}_1 + \theta\mathbf{d}_{x_1}) = \mathbf{b}_1$ and $A(\mathbf{x}_1 + \theta\mathbf{d}_{x_2}) = \mathbf{b}_1$, which imply that

$$\mathbf{d}_{B,x_1} = -A_{B1}^{-1}\mathbf{A}_{x_1} \text{ and } \mathbf{d}_{B,x_2} = -A_{B1}^{-1}\mathbf{A}_{x_2}.$$

The basic directions therefore are

$$\mathbf{d}_{x_1} = \begin{pmatrix} 1 \\ 0 \\ -A_{B1}^{-1}\mathbf{A}_{x_1} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ -1 \\ -1 \end{pmatrix} \text{ and } \mathbf{d}_{x_2} = \begin{pmatrix} 0 \\ 1 \\ -A_{B1}^{-1}\mathbf{A}_{x_2} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ -1 \\ -1 \\ 0 \end{pmatrix}.$$

By stepping in these directions, the changes in objective values are

$$\Delta c_{x_1} = \mathbf{c}^T(\theta\mathbf{d}_{x_1}) = \theta(c_{x_1} - \mathbf{c}_{B1}^T A_{B1}^{-1}\mathbf{A}_{x_1}) = \theta c_{x_1} = 0,$$

and

$$\Delta c_{x_2} = \mathbf{c}^T(\theta\mathbf{d}_{x_2}) = \theta(c_{x_2} - \mathbf{c}_{B1}^T A_{B1}^{-1}\mathbf{A}_{x_2}) = \theta c_{x_2} = -\theta,$$

respectively. In general, since $\theta > 0$, the objective value will decrease if

$$\bar{c}_{x_1} = c_{x_1} - \mathbf{c}_{B1}^T A_{B1}^{-1} \mathbf{A}_{x_1} < 0 \text{ or } \bar{c}_{x_2} = c_{x_2} - \mathbf{c}_{B1}^T A_{B1}^{-1} \mathbf{A}_{x_2} < 0,$$

where $\{\bar{c}_{x_1}, \bar{c}_{x_2}\}$ are the reduced costs of nonbasic variables $\{x_1, x_2\}$, which can be stacked in a vector $\bar{\mathbf{c}}_1 = \begin{pmatrix}\bar{c}_{x_1} & \bar{c}_{x_2}\end{pmatrix}^T$. The reduced cost of a nonbasic variable therefore quantifies the rate of change in objective value when stepping in the basic direction corresponding to this variable. As a result, if a basis is primal feasible and the reduced costs of all nonbasic variables are nonnegative, the basis is optimal. In the case at hand, the only option allowing us to reduce the objective value is thus to bring x_2 into the basis. The chosen basic direction is therefore $\mathbf{d}_1 = \mathbf{d}_{x_1}$ and the basic variable exiting the basis can be identified by inspecting the entries of

$$\mathbf{x}_2 = \mathbf{x}_1 + \theta_1 \mathbf{d}_1 = \begin{pmatrix}0\\0\\2\\6\\4\end{pmatrix} + \theta_1 \begin{pmatrix}0\\1\\-1\\-1\\0\end{pmatrix} = \begin{pmatrix}0\\\theta_1\\2-\theta_1\\6-\theta_1\\4\end{pmatrix} \geq \begin{pmatrix}0\\0\\0\\0\\0\end{pmatrix},$$

from which it is clear that $\theta_1 = 2$ is the maximum allowable step size, and s_1 is the first variable whose nonnegativity constraint becomes tight. Hence, it exits the basis. The updated basis is $\{x_2, s_2, s_3\}$, which corresponds to point B in Figure 12.1, as can be seen from the updated solution

$$\mathbf{x}_2 = \begin{pmatrix}0 & 2 & 0 & 4 & 4\end{pmatrix}^T.$$

In fact, these developments allow us to formulate a systematic criterion to identify the variable which should leave the basis. Indeed, it is worth remarking that $\mathbf{d}_{x_1} = -A_{B1}^{-1}\mathbf{A}_{x_1} = -\mathbf{A}_{x_1}$ and $\mathbf{d}_{x_2} = -A_{B1}^{-1}\mathbf{A}_{x_2} = -\mathbf{A}_{x_2}$. In other words, the entries of $\mathbf{d}_{x_1}$ and $\mathbf{d}_{x_2}$ are the opposite of the entries forming the columns of nonbasic variables in the simplex tableau. This is no fluke. In general, if a basis with basic matrix A_B is selected and the simplex tableau is brought to a form where the columns of basic variables form a permutation matrix, the columns of nonbasic variables will be obtained as $A_B^{-1}A_N$, with A_N the nonbasic matrix. Hence, the simplex tableau stores all of the necessary information to identify the variable that should exit the basis at any given time. Indeed, if the column corresponding to the nonbasic variable that should enter the basis has only negative entries, the problem is unbounded, since the objective value can be decreased without ever violating nonnegativity constraints. However, if this column has positive entries, the first basic variable for which the nonnegativity constraint will become tight will be the one for which the ratio between its value, which is shown in the last column of the tableau, and its coefficient in the tableau is smallest. Let $\mathcal{I} \subset \mathbb{N}$ and $\mathcal{J} \subset \mathbb{N}$ be two sets indexing constraints and variables, respectively. Then, for a nonbasic variable indexed by $j \in \mathcal{J}$ that should enter the basis, the corresponding column in the tableau has entries $\bar{a}_{ij},\ i \in \mathcal{I}$. In addition, let $\bar{b}_i,\ i \in \mathcal{I}$, denote the value of basic variables, as given by the last column in the simplex tableau. Then, the criterion specifying the (index of the) variable which should exit the basis can be expressed as

$$i_N = \arg\min \left\{ \frac{\bar{b}_i}{\bar{a}_{ij}} \middle| i \in \mathcal{I},\ \bar{a}_{ij} > 0 \right\}.$$

In this case, it is clear that s_1 should leave the basis rather than s_2, as $2/1 < 6/1$. This observation confirms the result obtained earlier, namely that the new basis should be $\{x_2, s_2, s_3\}$. Now, the simplex tableau must be updated accordingly. More precisely, the coefficients of all variables, their reduced costs and the values of basic variables must be updated. First, the updated basic matrix A_{B2} can be formed, along with its inverse A_{B2}^{-1},

$$A_{B2} = \begin{pmatrix}1 & 0 & 0\\1 & 1 & 0\\0 & 0 & 1\end{pmatrix} \text{ and } A_{B2}^{-1} = \begin{pmatrix}1 & 0 & 0\\-1 & 1 & 0\\0 & 0 & 1\end{pmatrix}.$$

Multiplying the columns of the previous tableau by the inverse of the basic matrix produces the updated columns. Indeed, by doing so, columns corresponding to the updated basic variables form a permutation matrix, while nonbasic columns are obtained via the multiplication of the nonbasic matrix A_{N2} by A_{B2}^{-1},

$$A_{N2} = \begin{pmatrix}-1 & 1\\1 & 0\\1 & 0\end{pmatrix} \text{ and } A_{B2}^{-1}A_{N2} = \begin{pmatrix}-1 & 1\\1 & 0\\1 & 0\end{pmatrix} = \begin{pmatrix}-1 & 1\\2 & -1\\1 & 0\end{pmatrix}.$$

Then, the cost vectors can be updated as $\mathbf{c}_{B2} = \begin{pmatrix}-1 & 0 & 0\end{pmatrix}^T$ and $\mathbf{c}_{N2} = \begin{pmatrix}0 & 0\end{pmatrix}^T$, such that the updated reduced costs of nonbasic variables become

$$\bar{\mathbf{c}}_{N2}^T = \mathbf{c}_{N2}^T - \mathbf{c}_{B2}^T A_{B2}^{-1} A_{N2} = \begin{pmatrix}-1 & 1\end{pmatrix}.$$

Finally, the values of the new basic variables are

$$\mathbf{x}_{B2} = A_{B2}^{-1}\mathbf{b}_1 = \begin{pmatrix}2 & 4 & 4\end{pmatrix}^T,$$

and the corresponding solution is

$$\mathbf{x}_2 = \begin{pmatrix}0 & 2 & 0 & 4 & 4\end{pmatrix}^T,$$

which confirms the developments made earlier. The resulting tableau therefore is

x_1	x_2	s_1	s_2	s_3	
-1	0	1	0	0	
-1	1	1	0	0	2
2	0	-1	1	0	4
1	0	0	0	1	4

from which it is clear that the only nonbasic variable with a negative reduced cost is x_1, which implies that it should enter the basis. Applying the criterion formulated earlier suggests that s_2 should leave the basis, as $4/2 < 4/1$. Thus, the new basis is $\{x_2, x_1, s_3\}$, while the nonbasic variables are $\{s_1, s_2\}$. Geometrically speaking, this basis corresponds to point C in Figure 12.1. Now, the simplex tableau should be updated accordingly. Previously, matrix operations were used to do so. A different, yet equivalent, approach to update the tableau consists in performing elementary row operations directly in the tableau, with the aim of i) setting the reduced cost of the new basic variable to 0 ii) ensuring that the columns of basic variables form a permutation matrix. In the case at hand, these row operations are i) $r_0 \leftarrow r_0 + \frac{1}{2}r_2$ ii) $r_1 \leftarrow r_1 + \frac{1}{2}r_2$ iii) $r_2 \leftarrow \frac{1}{2}r_2$ iv) $r_3 \leftarrow r_3 - \frac{1}{2}r_2$, and the resulting tableau is

x_1	x_2	s_1	s_2	s_3	
0	0	1/2	1/2	0	
0	1	1/2	1/2	0	4
1	0	-1/2	1/2	0	2
0	0	1/2	-1/2	1	2

The corresponding solution is $\mathbf{x}_3 = \begin{pmatrix}2 & 4 & 0 & 0 & 2\end{pmatrix}^T$, which is clearly primal feasible. In addition, the reduced costs of nonbasic variables are all nonnegative. Hence, the optimality conditions alluded to earlier are satisfied, and the problem is solved. Comparing the coordinates of C in Figure 12.1 with the values of x_1 and x_2 in $\mathbf{x}_3$ confirms that the latter indeed is the optimal solution.

Solution to Problem 2

Figure 12.2 displays a schematic of the problem at hand. In Figure 12.2, black lines represent the sets of points satisfying inequality constraints with equality, while blue arrows indicate the feasible half-space for each inequality constraint. The blue line corresponds to the only equality constraint, while the feasible set is shown in grey. In order to identify bases and their corresponding basic feasible solutions, the constraints should be expressed in standard form,

$$\begin{array}{rcrcrcrcr} 2x_1 & + & 3x_2 & & & & & = & 7 \\ x_1 & - & x_2 & + & s_1 & & & = & 0 \\ 3x_1 & + & 2x_2 & & & - & s_2 & = & 2 \\ x_1 & , & x_2 & , & s_1 & , & s_2 & \geq & 0. \end{array}$$

There are therefore 3 equality constraints and 4 variables. Recall that a basis can be formed by selecting as many variables as there are constraints, while making sure that the corresponding columns are linearly independent. These variables are then called *basic variables*, while all other remaining variables are referred to as *nonbasic variables*. When working with problems in standard form, basic variables usually have nonzero values, while nonbasic variables always have zero values. Intuitively, a basis characterises

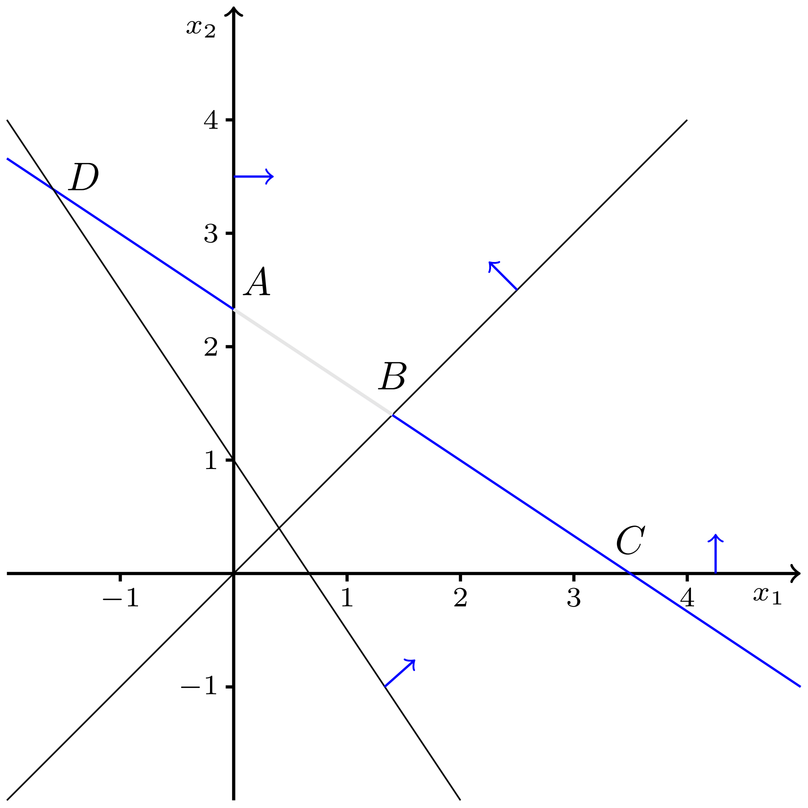


Figure 12.2: Schematic of Problem 2.

a solution to the system of equations formed by the equality constraints, and it may be sketched as a point in Figure 12.2. In particular, in Figure 12.2, A, B, C and D all correspond to different bases. From Figure 12.2, it is clear that a basis may be feasible, if the corresponding solution also satisfies nonnegativity constraints, or not. For instance, A corresponds to a feasible basis. Since $x_1 = 0$ at A, the basis is formed by $\{x_2, s_1, s_2\}$. Likewise, B corresponds to a feasible basis. At B, the second constraint is tight, i.e., $x_1 - x_2 = 0$, suggesting that $s_1 = 0$. Hence, the corresponding basis is $\{x_1, x_2, s_2\}$. By contrast, C and D correspond to bases which are not feasible. Indeed, at C, $x_2 = 0$ and the basis is $\{x_1, s_1, s_2\}$. However, $x_1 > 0$, such that $x_1 - x_2 > 0$ and the first inequality constraint is violated. Similarly, at D, the second inequality constraint is tight, such that $s_2 = 0$ and the basis is $\{x_1, x_2, s_1\}$. At D, $x_1 < 0$, which violates the nonnegativity constraint $x_1 \geq 0$. The basic feasible solutions corresponding to A and B can be obtained via a simplified simplex tableau. The initial tableau reads

x_1	x_2	s_1	s_2	
2	3	0	0	7
1	-1	1	0	0
3	2	0	-1	2

For A, the basis is $\{x_2, s_1, s_2\}$ and performing the following elementary row operations i) $r_1 \leftarrow \frac{1}{3}r_1$ ii) $r_2 \leftarrow r_2 + \frac{1}{3}r_1$ iii) $r_3 \leftarrow r_3 - \frac{2}{3}r_1$ iv) $r_3 \leftarrow -r_3$, yields

x_1	x_2	s_1	s_2	
2/3	1	0	0	7/3
5/3	0	1	0	7/3
-5/3	0	0	1	8/3

Likewise, for B, the basis is $\{x_1, x_2, s_2\}$, and the corresponding basic matrix A_B, along with its inverse A_B^{-1} are

$$A_B = \begin{pmatrix} 2 & 3 & 0 \\ 1 & -1 & 0 \\ 3 & 2 & -1 \end{pmatrix} \text{ and } A_B^{-1} = \begin{pmatrix} 1/5 & 3/5 & 0 \\ 1/5 & -2/5 & 0 \\ 1 & 1 & -1 \end{pmatrix}.$$

Applying the latter to the columns of the initial tableau leads to

x_1	x_2	s_1	s_2	
1	0	3/5	0	7/5
0	1	-2/5	0	7/5
0	0	1	1	5

Solution to Problem 3

A simple strategy to build the optimal tableau consists in identifying the optimal basis, whose basic matrix can be deduced, and then used to transform the initial tableau into the final, optimal tableau. Thus, the first step is to write the optimization problem in standard form,

$$\begin{array}{rrcrcrcrcrcrcrcrcrcrl} - & \min & -4x_1 & - & 7x_2 & - & 3x_3 & - & x_4 & - & 5x_5 & & & & & & & & \\ & \text{s.t.} & 2x_1 & + & 3x_2 & + & x_3 & & & + & 3x_5 & + & s_1 & & & & & = & 19, \\ & & x_1 & - & x_2 & & & + & 2x_4 & & & & & - & s_2 & & & = & 0, \\ & & x_1 & + & 3x_2 & & & + & x_4 & + & 2x_5 & & & & & + & s_3 & = & 10, \\ & & x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & , & s_1 & , & s_2 & , & s_3 & \geq & 0. \end{array}$$

from which the initial tableau can be constructed

x_1	x_2	x_3	x_4	x_5	s_1	s_2	s_3	
-4	-7	-3	-1	-5	0	0	0	
2	3	1	0	3	1	0	0	19
1	-1	0	2	0	0	-1	0	0
1	3	0	1	2	0	0	1	10

Inspecting the optimal solution reveals that x_3 and x_4 should be in the basis, while x_1, x_2 and x_5 should be nonbasic. Since there are three equality constraints, one of the three slack variables s_1, s_2 and s_3 should also be in the basis. Substituting the optimal solution in the equality constraints allows us to identify the values of the slack variables at optimality, from which the missing basic variable can be identified. Hence,

$$\begin{aligned} 0 + 0 + 19 + 0 + s_1 = 19 &\Rightarrow s_1 = 0, \\ 0 + 0 + 2 \times 10 - s_2 = 0 &\Rightarrow s_2 = 20, \\ 0 + 0 + 10 + 0 + s_3 = 10 &\Rightarrow s_3 = 0, \end{aligned}$$

from which it can be concluded that the optimal basis is $\{x_3, s_2, x_4\}$. The corresponding basic matrix A_B, nonbasic matrix A_N and the inverse of the former A_B^{-1} are

$$A_B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 2 \\ 0 & 0 & 1 \end{pmatrix}, \; A_N = \begin{pmatrix} 2 & 3 & 3 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 1 & 3 & 2 & 0 & 1 \end{pmatrix} \text{ and } A_B^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 2 \\ 0 & 0 & 1 \end{pmatrix},$$

from which the coefficients of nonbasic variables in the optimal tableau can be obtained as

$$A_B^{-1} A_N = \begin{pmatrix} 2 & 3 & 3 & 1 & 0 \\ 1 & 7 & 4 & 0 & 2 \\ 1 & 3 & 2 & 0 & 1 \end{pmatrix}.$$

The basic and nonbasic costs are

$$\mathbf{c}_B = \begin{pmatrix} -3 & 0 & -1 \end{pmatrix}^T \text{ and } \mathbf{c}_N = \begin{pmatrix} -4 & -7 & -5 & 0 & 0 \end{pmatrix}^T,$$

from which the reduced costs of nonbasic variables can be computed

$$\bar{\mathbf{c}}_N^T = \mathbf{c}_N^T - \mathbf{c}_B^T A_B^{-1} A_N = \begin{pmatrix} 3 & 5 & 6 & 3 & 1 \end{pmatrix}.$$

It is worth noticing that all reduced costs are nonnegative, which confirms that the basis is optimal. It is also straightforward to check that multiplying the last column of the initial tableau by the inverse of the basic matrix indeed yields the optimal solution. The optimal tableau therefore reads

x_1	x_2	x_3	x_4	x_5	s_1	s_2	s_3	
3	5	0	0	6	3	0	1	
2	3	1	0	3	1	0	0	19
1	7	0	0	4	0	1	2	20
1	3	0	1	2	0	0	1	10

This result could have also been reached by performing the following elementary row operations on the initial tableau, i) $r_0 \leftarrow r_0 + 3r_1$ ii) $r_0 \leftarrow r_0 + r_3$ iii) $r_2 \leftarrow -r_2$, and iv) $r_2 \leftarrow r_2 + 2r_3$. Indeed, it is worth remarking that

$$-\mathbf{c}_B^T A_B^{-1} = \begin{pmatrix} 3 & 0 & 1 \end{pmatrix}$$

encodes operations i) and ii), while the inverse of the basic matrix A_B^{-1} encodes operations iii) and iv).

Solution to Problem 4

1. The problem has three constraints and their right-hand side coefficients are all nonnegative. Hence, adding three auxiliary variables $x_6 \in \mathbb{R}_+, x_7 \in \mathbb{R}_+, x_8 \in \mathbb{R}_+$ is therefore sufficient for our purpose. Let M be the positive constant used to penalise the auxiliary variables. The *Big M* problem in standard form therefore reads

$$\begin{array}{lrrrrrrrrrrrrrrrrl}
\min & & - & x_2 & & & & & & & + & Mx_6 & + & Mx_7 & + & Mx_8 & & \\
\text{s.t.} & x_1 & + & x_2 & - & x_3 & & & & & + & x_6 & & & & & = & 1 \\
& 2x_1 & - & x_2 & & & - & x_4 & & & & & + & x_7 & & & = & 2 \\
& 2x_1 & + & x_2 & & & & & + & x_5 & & & & & + & x_8 & = & 6 \\
& x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & , & x_6 & , & x_7 & , & x_8 & \geq & 0
\end{array}$$

2. Based on the previous developments, the *phase I* linear program reads

$$\begin{array}{lrrrrrrrrrrrrrrrrl}
\min & & & & & & & & & & & x_6 & + & x_7 & + & x_8 & & \\
\text{s.t.} & x_1 & + & x_2 & - & x_3 & & & & & + & x_6 & & & & & = & 1 \\
& 2x_1 & - & x_2 & & & - & x_4 & & & & & + & x_7 & & & = & 2 \\
& 2x_1 & + & x_2 & & & & & + & x_5 & & & & & + & x_8 & = & 6 \\
& x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & , & x_6 & , & x_7 & , & x_8 & \geq & 0
\end{array}$$

Note that the only difference between the *Big M* and *phase I* problems is the objective function used. Moreover, the latter is simpler and more robust, and it is therefore the preferred method.

In order to solve the *phase I* problem, a starting basic feasible solution can be constructed by taking $\{x_6, x_7, x_8\}$ as basic variables. The associated simplex tableau reads

−5	−1	1	1	−1	0	0	0	
1	1	−1	0	0	1	0	0	1
2	−1	0	−1	0	0	1	0	2
2	1	0	0	1	0	0	1	6

where the reduced costs were computed as follows

$$\bar{\mathbf{c}}^T = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}^T - \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 2 & -1 & 0 & -1 & 0 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Now, let us assume that x_1 enters and x_6 leaves the basis, respectively. We perform the following operations: i) $r_2 \leftarrow r_2 + 2r_1$, ii) $r_3 \leftarrow r_3 - 2r_1$, iii) $c \leftarrow c + 5r_1$, such that the updated tableau reads

0	4	−4	1	−1	5	0	0	
1	1	−1	0	0	1	0	0	1
0	−3	2	−1	0	−2	1	0	0
0	−1	2	0	1	−2	0	1	4

After several iterations of the primal simplex algorithm, we obtain a solution $x^*_{phase\ I} = \{3, 0, 2, 4, 0, 0, 0, 0\}$. Clearly, the auxiliary variables are nonbasic in this solution. Hence, we conclude that the original problem has a non-empty solution set and that the basis $\{x_1, x_3, x_4\}$ is primal feasible for the original problem. The *phase II* problem can then be constructed from the *phase I* optimal tableau, where reduced costs must be updated accordingly.

Chapter 13

The Revised Simplex, LP Duality and the Dual Simplex

Solution to Problem 1

1. Let $A \in \mathbb{R}^{m\times n}$ denote the coefficient matrix, with $m = 5$ constraints and $n = 10$ variables. Recall that once a basis B has been selected, a partition of the coefficient matrix can be produced, i.e., two submatrices $A_B \in \mathbb{R}^{m\times m}$ and $A_N \in \mathbb{R}^{m\times(n-m)}$ can be formed such that $A = \begin{pmatrix} A_B & A_N \end{pmatrix}$, possibly after permuting some columns. Let $x \in \mathbb{R}^n$ denote the basic feasible solution associated with this basis, with entries $x^T = \begin{pmatrix} x_B^T & x_N^T \end{pmatrix}$, and let $b \in \mathbb{R}^m$ be the vector of right-hand side coefficients. The equality constraints can thus be expressed compactly as

$$Ax = b \Leftrightarrow \begin{pmatrix} A_B & A_N \end{pmatrix} \begin{pmatrix} x_B \\ x_N \end{pmatrix} = b \Leftrightarrow A_B x_B + A_N x_N = b.$$

In addition, since nonbasic variables are equal to 0, $x_N = 0$, and

$$A_B x_B = b.$$

By definition of a basis, A_B is nonsingular, and the values of variables forming this basis can be readily obtained as

$$x_B = A_B^{-1} b = \bar{b}.$$

In the case at hand, let $B = \{x_1, x_2, s_3, s_4, s_5\}$ denote the current basis. The inverse of the basis matrix A_B^{-1} is known, while $b = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \end{pmatrix}^T$, thus

$$x_B = \bar{b} = \begin{pmatrix} 1 & 0 & 2 & 4 & 5 \end{pmatrix}^T,$$

or $x_1 = 1$, $x_2 = 0$, $s_3 = 2$, $s_4 = 4$ and $s_5 = 5$. It is worth noting that the solution is degenerate, as $x_2 = 0$, which implies that the associated nonnegativity constraint is tight.

2. Let $c \in \mathbb{R}^n$ be the cost vector, which can also be partitioned into $c^T = \begin{pmatrix} c_B^T & c_N^T \end{pmatrix}$, with

$$\begin{aligned} c_B^T &= \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \end{pmatrix} \\ c_N^T &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}. \end{aligned}$$

In addition, let $p \in \mathbb{R}^m$ be such that $p = (A_B^T)^{-1} c_B = (A_B^{-1})^T c_B = \begin{pmatrix} 3 & -1 & 0 & 0 & 0 \end{pmatrix}^T$. Note that p can be interpreted as dual variables. Indeed, for an optimal basis $B^\star$, strong duality implies

$$c_{B^\star}^T x_{B^\star} = c^T x^\star = (p^\star)^T b = (p^\star)^T Ax = (p^\star)^T A_{B^\star} x_{B^\star},$$

from which the expression of p given above can be recovered. Now, let $\bar{c} \in \mathbb{R}^n$ denote the reduced costs vector, which can be partitioned into $\bar{c}^T = \begin{pmatrix} \bar{c}_B^T & \bar{c}_N^T \end{pmatrix}$. Then, by definition, the reduced costs can be obtained as $\bar{c}^T = c^T - p^T A$. Clearly, $\bar{c}_B^T = c_B^T - p^T A_B = c_B^T - c_B^T A_B^{-1} A_B = 0$, while $\bar{c}_N^T = c_N^T - p^T A_N$, which yields

$$\bar{c}_N^T = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}^T - \begin{pmatrix} 3 & -1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \\ 1 \\ -3 \\ 1 \end{pmatrix}^T ,$$

In other words, $\bar{c}_{x_3} = 4$, $\bar{c}_{x_4} = 0$, $\bar{c}_{x_5} = 1$, $\bar{c}_{s_1} = -3$ and $\bar{c}_{s_2} = 1$, from which it is deduced that s_1 should enter the basis.

3. Recall that nonbasic columns result from elementary row operations performed in the tableau so as to form a permutation matrix with basic columns. Furthermore, note that the inverse basis matrix encodes these elementary row operations. In particular, let $A_{s_1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}^T$ be the column of s_1 in the initial tableau. In the case at hand, the updated basis is $\{x_1, x_2, s_3, s_4, s_5\}$, and the matrix A_B^{-1} making it possible to move from the initial tableau to the current basis is known. Let $\bar{A}_{s_1}$ denote the column associated with s_1 that would appear in the current tableau if it was maintained, which can be computed as

$$\bar{A}_{s_1} = A_B^{-1} A_{s_1} = \begin{pmatrix} 1 & -2 & 1 & 0 & 0 \end{pmatrix}^T .$$

In order to identify the variable which should exit the basis, the ratio test can be invoked. Recall that applying this criterion is equivalent to i) extracting the positive entries in the column corresponding to the nonbasic variable that will enter the basis ii) extracting the right-hand side coefficients corresponding to these entries from the last column of the tableau iii) computing the ratio of right-hand side coefficients to column coefficients iv) keeping the index yielding the minimum ratio value. This index then indicates the variable which should exit the basis. In this case, either x_1 or s_3 should exit the basis. As $1/1 < 2/1$, x_1 exits the basis.

4. Recall that in the standard primal simplex algorithm studied in Problem Set 2, a simplex tableau was maintained and updated at each iteration. Moreover, each basis update (or pivot) involved elementary row operations, or equivalently, multiplying the tableau by the inverse of the basic matrix. Unfortunately, performing a large number of pivots may destroy the sparsity of the initial tableau that is typical of carefully formulated linear programs, resulting in a tableau with very few nonzero coefficients. This phenomenon, which is known as *fill-in*, leads to less computationally efficient pivots and increased memory requirements as the number of iterations grows. The revised simplex algorithm takes advantage of the fact that the inverse basis matrix encodes the elementary row operations needed to perform a pivot and attempts to alleviate the fill-in problem by storing and working directly on the inverse basis matrix at each iteration. In this setup, the stored inverse matrix can be viewed as a product (composition) of inverse matrices from all previous pivots. Alternatively, it can be understood as a product of elementary matrices implementing all the necessary elementary row operations to move from the initial tableau to the current one.

In this case, x_1 is exiting the basis B and s_1 is entering it. Let $B' = \{s_1, x_2, s_3, s_4, s_5\}$ be the new basis. Since x_1 is the first basic variable in B, the elementary row operations implementing this pivot aim to replace the nonbasic column $\bar{A}_{s_1}$ by the first column of an $m \times m$ identity matrix. It is straightforward to see that these operations are i) $r_2 \leftarrow r_2 + 2r_1$ ii) $r_3 \leftarrow r_3 - r_1$ iii) $r_i \leftarrow r_i,\ i = 1, 4, 5$. These operations are encoded in the following matrix

$$A_{B'}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} .$$

It is no coincidence that this matrix is precisely the inverse of the updated basis matrix

$$A_{B'} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Building upon these developments, the matrix P allowing one to move from the initial simplex tableau to the new one can be written as the product of $A_{B'}^{-1}$ and A_B^{-1}

$$P = A_{B'}^{-1}A_B^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Applying the aforementioned elementary row operations on A_B^{-1} would have yielded the same result. The values of basic variables can be simply updated as

$$x_{B'} = A_{B'}^{-1}\bar{b} = Pb = \begin{pmatrix} 1 & 2 & 1 & 4 & 5 \end{pmatrix}^T,$$

and the associated objective value is $c_{B'}^T x_{B'} = 0 - 2 + 0 = -2 \leq c_B^T x_B = 1 + 0 + 0 = 1$.

More generally, for a sequence of pivots represented by matrices $P_k \in \mathbb{R}^{m\times m}, k = 1, \dots, K$, the transformation allowing one to move from the initial tableau to the current tableau would be written as $P = P_K P_{K-1} ... P_2 P_1$. Note that each matrix $P_k, k = 1, \dots K$, has at most $2m - 1$ nonzero entries, and is therefore sparse. Hence, storing the factors of P can be more efficient than storing P itself as it may become fairly dense after a large number of pivots. However, some overhead comes from the need to recompute P from its factors at each iteration. Finally, it is worth mentioning that in modern implementations of the primal simplex algorithm, the LU factorisation of the basis matrix is kept in memory and updated at each pivot, instead of working on an inverse basis matrix or a sequence of inverse basis matrices. Some of the most common LU update rules include the Bartels-Golub, the Forrest-Tomlin, the Suhl-Suhl, the Fletcher-Matthews and Schur complement-based rules.

Solution to Problem 2

Solution 1

Let $p_1 \in \mathbb{R}_-$, $p_2 \in \mathbb{R}_+$ and $p_3 \in \mathbb{R}$ be dual variables associated with the first, second and third constraints of the primal problem, respectively,

$$\begin{aligned} \min \ & x_1 - x_2 \\ \text{s.t. } & 2x_1 + 3x_2 - x_3 + x_4 \leq 0 : p_1, \\ & 3x_1 + x_2 + 4x_3 - 2x_4 \geq 3 : p_2, \\ & -x_1 - x_2 + 2x_3 + x_4 = 6 : p_3, \\ & x_1 \in \mathbb{R}_-, \ x_2 \in \mathbb{R}_+, \ x_3 \in \mathbb{R}_+, \ x_4 \in \mathbb{R}. \end{aligned}$$

Dualising these constraints yields the following problem

$$\begin{aligned} d(p_1, p_2, p_3) = \min \ & x_1 - x_2 + p_1(-2x_1 - 3x_2 + x_3 - x_4) + p_2(3 - 3x_1 - x_2 - 4x_3 + 2x_4) + \\ & p_3(6 + x_1 + x_2 - 2x_3 - x_4) \\ \text{s.t. } & x_1 \in \mathbb{R}_-, \ x_2 \in \mathbb{R}_+, \ x_3 \in \mathbb{R}_+, \ x_4 \in \mathbb{R}. \end{aligned}$$

Note that the range of the dual variables was selected to guarantee that $d(p_1, p_2, p_3)$ provides a lower bound on the primal objective while promoting primal feasibility. Then, after rearranging terms in the objective function, the problem becomes

$$d(p_1, p_2, p_3) = \min \quad (1 - 2p_1 - 3p_2 + p_3)x_1 + (-1 - 3p_1 - p_2 + p_3)x_2 + (p_1 - 4p_2 - 2p_3)x_3 +$$

$$
\begin{aligned}
&(-p_1 + 2p_2 - p_3)x_4 + 3p_2 + 6p_3 \\
\text{s.t. } & x_1 \in \mathbb{R}_-,\ x_2 \in \mathbb{R}_+,\ x_3 \in \mathbb{R}_+,\ x_4 \in \mathbb{R}.
\end{aligned}
$$

Inspecting the first four terms reveals that the following conditions must be satisfied to guarantee that the relaxed problem above is bounded,

$$
\begin{aligned}
1 - 2p_1 - 3p_2 + p_3 &\leq 0 \\
-1 - 3p_1 - p_2 + p_3 &\geq 0 \\
p_1 - 4p_2 - 2p_3 &\geq 0 \\
-p_1 + 2p_2 - p_3 &= 0,
\end{aligned}
$$

in which case $d(p_1, p_2, p_3) = 3p_2 + 6p_3$. Seeking the tightest lower bound on the primal objective therefore leads to the dual problem

$$
\begin{aligned}
\max \quad & 3p_2 + 6p_3 \\
\text{s.t. } & 1 - 2p_1 - 3p_2 + p_3 \leq 0 \\
& -1 - 3p_1 - p_2 + p_3 \geq 0 \\
& p_1 - 4p_2 - 2p_3 \geq 0 \\
& -p_1 + 2p_2 - p_3 = 0 \\
& p_1 \in \mathbb{R}_-,\ p_2 \in \mathbb{R}_+,\ p_3 \in \mathbb{R},
\end{aligned}
$$

or, equivalently,

$$
\begin{aligned}
\max \quad & 3p_2 + 6p_3 \\
\text{s.t. } & 2p_1 + 3p_2 - p_3 \geq 1 : x_1 \\
& 3p_1 + p_2 - p_3 \leq -1 : x_2 \\
& -p_1 + 4p_2 + 2p_3 \leq 0 : x_3 \\
& p_1 - 2p_2 + p_3 = 0 : x_4 \\
& p_1 \in \mathbb{R}_-,\ p_2 \in \mathbb{R}_+,\ p_3 \in \mathbb{R}.
\end{aligned}
$$

Now, let $x = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \end{pmatrix}^T$, $p = \begin{pmatrix} p_1 & p_2 & p_3 \end{pmatrix}^T$,

$$
c = \begin{pmatrix} c_1 & c_2 & c_3 & c_4 \end{pmatrix}^T = \begin{pmatrix} 1 & -1 & 0 & 0 \end{pmatrix}^T,\ b = \begin{pmatrix} b_1 & b_2 & b_3 \end{pmatrix}^T = \begin{pmatrix} 0 & 3 & 6 \end{pmatrix}^T,
$$

and

$$
A = \begin{pmatrix} a_1^T \\ a_2^T \\ a_3^T \end{pmatrix} = \begin{pmatrix} 2 & 3 & -1 & 1 \\ 3 & 1 & 4 & -2 \\ -1 & -1 & 2 & 1 \end{pmatrix} = \begin{pmatrix} A_1 & A_2 & A_3 & A_4 \end{pmatrix}.
$$

Then, the primal problem can be expressed in matrix form

$$
\begin{aligned}
\min \quad & c^T x \\
\text{s.t. } & a_1^T x \leq b_1 : p_1 \\
& a_2^T x \geq b_2 : p_2 \\
& a_3^T x = b_3 : p_3 \\
& x \in \mathbb{R}_- \times \mathbb{R}_+^2 \times \mathbb{R},
\end{aligned}
$$

and the dual problem reads

$$
\begin{aligned}
\max \quad & b^T p \\
\text{s.t. } & A_1^T p \geq c_1 : x_1 \\
& A_2^T p \leq c_2 : x_2 \\
& A_3^T p \leq c_3 : x_3 \\
& A_4^T p = c_4 : x_4 \\
& p \in \mathbb{R}_- \times \mathbb{R}_+ \times \mathbb{R}.
\end{aligned}
$$

Solution 2

In the particular case of linear programming, a lower bound on the primal objective can be readily constructed by taking a linear combination of primal constraints. The coefficients of this linear combination, which correspond to dual variables, can then optimised to tighten the bound, yielding the dual problem.

More formally, let p_1, p_2 and p_3 be the dual variables. Their range will be discussed shortly. Linearly combining the terms on the left-hand side of the primal constraints yields

$$(2x_1+3x_2-x_3+x_4)p_1+(3x_1+x_2+4x_3-2x_4)p_2+(-x_1-x_2+2x_3+x_4)p_3, \tag{13.1}$$

while the linear combination of terms on the right-hand side gives

$$0p_1+3p_2+6p_3. \tag{13.2}$$

The key idea is to produce a sequence of inequalities, whereby the primal objective is lower bounded by the linear combination of left-hand side terms (13.1), which is itself lower bounded by the linear combination of right-hand side terms (13.2). Indeed, the latter only depends on p_1, p_2 and p_3 and can easily be maximised to tighten the bounds. In order to make sure that (13.1) indeed provides an upper bound on (13.2), the range of the dual variables must be carefully selected. In the case at hand, taking $p_1 \in \mathbb{R}_+$, $p_2 \in \mathbb{R}_-$ and $p_3 \in \mathbb{R}$ ensures that

$$(2x_1+3x_2-x_3+x_4)p_1+(3x_1+x_2+4x_3-2x_4)p_2+(-x_1-x_2+2x_3+x_4)p_3 \geq 0p_1+3p_2+6p_3,$$

for any feasible combination of primal variables. Next, conditions guaranteeing that the primal objective is lower bounded by (13.1) are sought. One successively finds

$$\begin{aligned} x_1-x_2+0x_3+0x_4 &\geq (2x_1+3x_2-x_3+x_4)p_1+(3x_1+x_2+4x_3-2x_4)p_2+(-x_1-x_2+2x_3+x_4)p_3 \\ &= (2p_1+3p_2-p_3)x_1+(3p_1+p_2-p_3)x_2+(-p_1+4p_2+2p_3)x_3+(p_1-2p_2+p_3)x_4. \end{aligned}$$

Given the range of the primal variables $x_1 \in \mathbb{R}_-$, $x_2 \in \mathbb{R}_+$, $x_3 \in \mathbb{R}_+$ $x_4 \in \mathbb{R}$, inspecting the coefficients of primal variables on both sides of the inequality reveals that it holds if

$$\begin{array}{rcrcrcr} 2p_1 & + & 3p_2 & - & p_3 & \geq & 1 \\ 3p_1 & + & 2p_2 & - & p_3 & \leq & -1 \\ -\ p_1 & + & 4p_2 & + & 2p_3 & \leq & 0 \\ p_1 & - & 2p_2 & + & p_3 & = & 0 \end{array}. \tag{13.3}$$

In order to obtain the tightest lower bound possible, (13.2) is maximized subject to constraints (13.3), which yields the dual problem

$$\begin{array}{lcrcrcrcr} \max & & & & 3p_2 & + & 6p_3 & & \\ \text{s.t.} & & 2p_1 & + & 3p_2 & - & p_3 & \geq & 1 \\ & & 3p_1 & + & 2p_2 & - & p_3 & \leq & -1 \\ & - & p_1 & + & 4p_2 & + & 2p_3 & \leq & 0 \\ & & p_1 & - & 2p_2 & + & p_3 & = & 0 \\ & & p_1 \in \mathbb{R}_+ & , & p_2 \in \mathbb{R}_- & , & p_3 \in \mathbb{R} & & \end{array} \tag{13.4}$$

It can be shown that at optimality, the lower bound is tight in the sense that the dual objective and the primal objective have the same value. Formally, let $x_i^\star, i=1,\dots,n$, and $p_j^\star, j=1,\dots,m$, denote optimal primal and dual variables, respectively. It follows that

$$x_1^\star - x_2^\star = 3p_2^\star + 6p_3^\star. \tag{13.5}$$

This property is called the strong duality of linear programs.

Solution to Problem 3

We start by checking that $x^\star$ is primal feasible,

$$-2x_1^\star + x_2^\star = -2+1 = -1 \leq -1$$

$$\begin{aligned}x_1^\star - 3x_2^\star &= 1 - 3 = -2 \leq -2\\ x_1^\star + x_2^\star &= 1 + 1 = 2 \leq 5,\end{aligned}$$

and all three inequality constraints are indeed satisfied. Note that the first two constraints are tight, i.e., they are satisfied with equality. Now, let

$$A = \begin{pmatrix} a_1^T \\ a_2^T \\ a_3^T \end{pmatrix} = \begin{pmatrix} -2 & 1 \\ 1 & -3 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} A_1 & A_2 \end{pmatrix},\ c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix},\ b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} -1 \\ -2 \\ 5 \end{pmatrix}.$$

In addition, let $p^\star = \begin{pmatrix} p_1^\star & p_2^\star & p_3^\star \end{pmatrix}^T$ denote the set of (candidate) optimal dual variables, with $p_i^\star \in \mathbb{R}_-$, $i = 1, \ldots, 3$. Recall that, at optimality, the complementary slackness conditions of the primal problem should be satisfied, that is,

$$p_i^\star(a_i^T x^\star - b_i) = 0,\ i = 1, \ldots, 3.$$

Likewise, for the dual problem, the complementary slackness conditions should be verified at optimality, such that

$$x_j^\star(c_j - A_j^T p^\star) = 0,\ j = 1, 2.$$

Since the third constraint is not tight, $a_3^T x^\star - b_3 = 2 - 5 = -3$ and $p_3^\star = 0$ to satisfy the complementary slackness conditions of the primal problem. Then, writing the dual complementary slackness conditions in full and substituting $p_3^\star = 0$ yields the following linear equations,

$$\begin{aligned}1 \times \Big(2 - (-2p_1^\star + 1p_2^\star)\Big) &= 0\\ 1 \times \Big(1 - (1p_1^\star - 3p_2^\star)\Big) &= 0.\end{aligned}$$

This system of linear equations can be rewritten as

$$\begin{pmatrix} -2 & 1 \\ 1 & -3 \end{pmatrix} \begin{pmatrix} p_1^\star \\ p_2^\star \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix},$$

whose unique solution is $p_1^\star = -7/5$ and $p_2^\star = -4/5$. Finally, recall that the strong duality property of linear programs implies that

$$c^T x^\star = b^T p^\star,$$

for the optimal primal-dual pair $(x^\star, p^\star)$. In the case at hand, $c^T x^\star = 3$ and $b^T p^\star = (-1) \times (-7/5) + (-2) \times (-4/5) + 5 \times 0 = 15/5 = 3$, such that $c^T x^\star = b^T p^\star$ and $x^\star$ is indeed an optimal solution to the primal problem.

Chapter 14

Sensitivity Analysis in Linear Programming

Solution to Problem 1

1. Since the final simplex tableau is given, the optimal basis can be readily identified. In the case at hand, the corresponding basic variables are x_3, x_2 and s_1, respectively. In order to retrieve the associated basic matrix, it is helpful to write the initial simplex tableau

x_1	x_2	x_3	s_1	s_2	s_3	b
-2	-5	-1	0	0	0	
1	0	1	-1	0	0	1
0	-1	2	0	-1	0	0
3	0	1	0	0	1	5

where it is worth noticing that i) the costs are negative, since the original problem is a maximization problem which has been expressed in standard form ii) the coefficients associated with the slack variables of the first two constraints are negative because these are $\geq$ inequality constraints. From the initial tableau, it is clear that the basic matrix is

$$A_B = \begin{pmatrix} 1 & 0 & -1 \\ 2 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix},$$

and it can be verified that its inverse is

$$A_B^{-1} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & 2 \\ -1 & 0 & 1 \end{pmatrix}.$$

Although only a factorization of A_B^{-1} is maintained and updated in the revised simplex algorithm, recall that this matrix encodes the elementary row operations that must be performed in the initial simplex tableau to obtain the final simplex tableau. The values of the basic variables can thus be computed as $x_B = A_B^{-1}b$, with

$$b = \begin{pmatrix} 1 \\ 0 \\ 5 \end{pmatrix} \text{ and } x_B = \begin{pmatrix} 5 \\ 10 \\ 4 \end{pmatrix}.$$

It is worth noticing that x_B also stores the entries of the last column of the final simplex tableau. Now, changing the right-hand side coefficient of the second inequality constraint is equivalent to introducing an updated right-hand side coefficients vector

$$b' = \begin{pmatrix} 1 \\ 2 \\ 5 \end{pmatrix}.$$

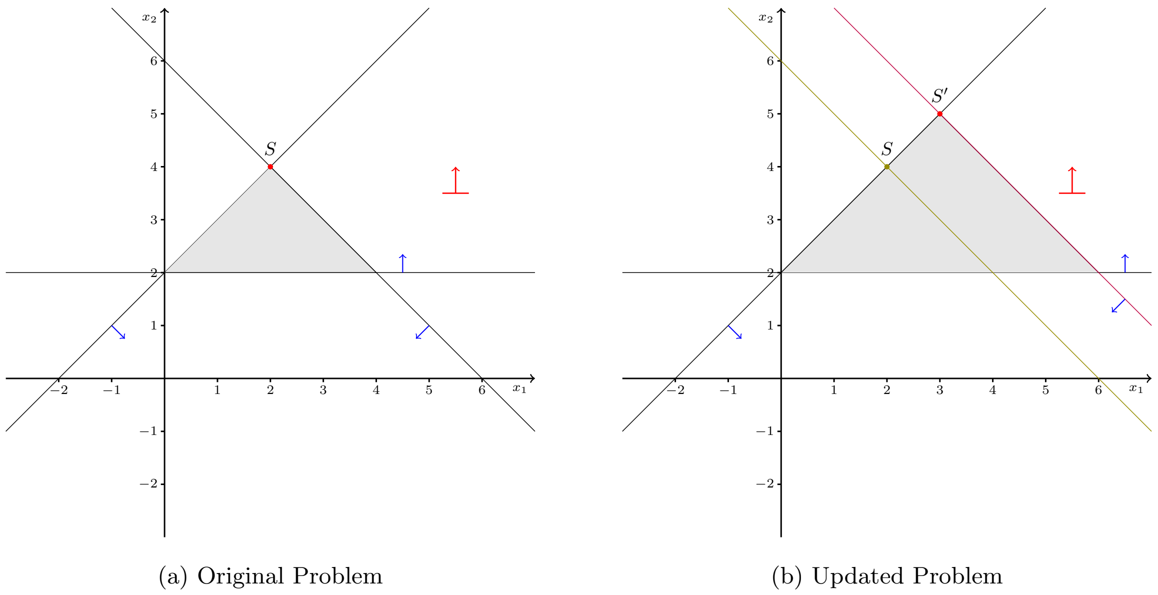


Figure 14.1: Schematic of right-hand side coefficient update and its impact on the optimal solution.

Provided that this update does not lead to a different optimal basis, applying the inverse basic matrix to the updated right-hand side coefficients vector will directly yield the updated values of the basic variables, that is,

$$x'_B = A_B^{-1}b' = \begin{pmatrix} 5 \\ 8 \\ 4 \end{pmatrix}.$$

Since $x'_B \geq 0$ and the reduced costs are not affected by a change in right-hand side coefficients, the basis remains optimal. A schematic of the situation is illustrated in Figure 14.1 for a simple problem. Figure 14.1a shows the original problem and its solution S, whereas Figure 14.1b displays the updated problem and the new optimal solution S'. The basis is unchanged, as the set of active constraints remains the same. By contrast, introducing

$$b'' = \begin{pmatrix} 1 \\ 0 \\ -3 \end{pmatrix},$$

leads to

$$x''_B = A_B^{-1}b'' = \begin{pmatrix} -3 \\ -6 \\ -4 \end{pmatrix},$$

which is clearly primal infeasible, as $x''_B < 0$. In such a case, dual simplex iterations should be performed to try and recover a new optimal basis or show that the resulting problem is infeasible.

2. Recall that the reduced costs of basic variables are always equal to 0, whereas the reduced costs of non-basic variables can be computed as $\bar{c}_N^T = c_N^T - c_B^T A_B^{-1} A_N$, with c_N and c_B the coefficients of non-basic and basic variables in the objective function, respectively, and A_N the coefficients of non-basic variables in the initial simplex tableau. It is worth remarking that $A_B^{-1}A_N$ also corresponds to the coefficients of non-basic variables in the final simplex tableau. Since x_3 is the first basic variable, updating the value of its coefficient in the objective function amounts to changing the first entry of $c_B^T = \begin{pmatrix} -1 & -5 & 0 \end{pmatrix}$. Let δ denote this new coefficient value, such that the updated cost vector $c_B'^T = \begin{pmatrix} \delta & -5 & 0 \end{pmatrix}$ is formed. This update will also lead to updated reduced costs values, which correspond to the entries of the reduced cost vector $\bar{c}_N'^T = c_N^T - c_B'^T A_B^{-1} A_N$. For the optimal basis to remain unchanged, the updated reduced costs should remain nonnegative, that is, $\bar{c}'_N \geq 0$. Hence, one inequality can be extracted for each entry of $\bar{c}'_N$

in order to identify the maximum value δ for which the basis remains unchanged. Since

$$c_N^T = \begin{pmatrix} -2 & 0 & 0 \end{pmatrix} \text{ and } A_B^{-1}A_N = \begin{pmatrix} 3 & 0 & 1 \\ 6 & 1 & 2 \\ 2 & 0 & 1 \end{pmatrix},$$

the updated reduced costs vector is $\bar{c}_N'^T = \begin{pmatrix} -3\delta + 28 & 5 & -\delta + 10 \end{pmatrix}$. The nonnegativity conditions imply that $\delta \leq 28/3$ and $\delta \leq 10$, from which it is clear that $\delta = 28/3$ is the largest value leaving the basis unchanged.

3. The initial tableau must be updated to include the new constraint and reads

x_1	x_2	x_3	s_1	s_2	s_3	s_4	b
-2	-5	-1	0	0	0	0	
1	0	1	-1	0	0	0	1
0	-1	2	0	-1	0	0	0
3	0	1	0	0	1	0	5
1	-1	0	0	0	0	-1	5

Given the previously optimal basis, a new basis B' can be formed for the updated problem by adding s_4 to this basis. The tableau must then be updated in order to check whether the new basis is also optimal for the updated problem. To this end, the new inverse matrix can be computed using the previous inverse basis matrix and the coefficients of the new constraint. Indeed, adding an inequality constraint with a surplus variable (negative slack variable) and coefficients a_{m+1}^T for the other variables is equivalent to considering a standard form linear program with updated coefficient matrix

$$A' = \begin{pmatrix} A & 0 \\ a_{m+1}^T & -1 \end{pmatrix}$$

and updated right-hand side coefficients

$$b' = \begin{pmatrix} b \\ b_{m+1} \end{pmatrix}.$$

Since s_4 is the additional basic variable, the updated basic matrix reads

$$A'_{B'} = \begin{pmatrix} A_B & 0 \\ a_B^T & -1 \end{pmatrix},$$

where $a_B^T = a_{m+1,B}^T$ for simplicity. Then, it is straightforward to check that the inverse of the updated basic matrix is

$$A_{B'}'^{-1} = \begin{pmatrix} A_B^{-1} & 0 \\ a_B^T A_B^{-1} & -1 \end{pmatrix},$$

such that the coefficients of the updated tableau are obtained as

$$\bar{A}' = A_{B'}'^{-1} A' = \begin{pmatrix} A_B^{-1}A & 0 \\ a_B^T A_B^{-1} A - a_{m+1}^T & 1 \end{pmatrix},$$

while the updated right-hand side coefficients can be expressed as

$$\bar{b}' = A_{B'}'^{-1} b' = \begin{pmatrix} A_B^{-1}b \\ a_B^T A_B^{-1} b - b_{m+1} \end{pmatrix}.$$

Clearly, all tableau rows and columns are the same as in the previous final tableau, except for the last row and column, which is are new and correspond to the surplus variable s_4 and the new constraint, respectively. Since the new column is trivially obtained, only the new row in the updated tableau must be computed, which can be achieved using the previous basic matrix and the coefficients of the new constraint. It is also worth mentioning that since the s_4 variable has a zero coefficient in the objective function, the reduced costs in the updated tableau are the same as those in the final tableau used previously. In addition, since s_4 belongs to the basis, its reduced cost remains equal to 0. The updated tableau therefore reads

x_1	x_2	x_3	s_1	s_2	s_3	s_4	b
31	0	0	0	5	11	0	
3	0	1	0	0	1	0	5
6	1	0	0	1	2	0	10
2	0	0	1	0	1	0	4
-7	0	0	0	-1	-2	1	-15

Unfortunately, it is clear from the last column that this basis is primal infeasible. Indeed, the s_4 variable has a negative value and therefore violates nonnegativity constraints. Nevertheless, this basis is dual feasible, as all reduced costs are nonnegative. Hence, performing an iteration of the dual simplex algorithm may yield a new basis with more favourable properties. The variable that will replace s_4 in the basis must be identified next. Let $\mathcal{I} \subset \mathbb{N}$ be a set indexing all variables. Recall that the condition to identify (the index of) the variable entering the basis can be expressed as

$$I = \arg\min \left\{ \frac{\bar{c}'_i}{|\bar{a}'_{ri}|} \middle| i \in \mathcal{I},\ \bar{a}'_{ri} < 0 \right\},$$

where r is the index of the row associated with the variable violating the nonnegativity constraint, $\bar{c}'_i$ denotes the i^{th} entry of the updated reduced cost vector, and $\bar{a}'_{ri}$ is the entry in row r and column i of $\bar{A}'$. In this case, $r = 4$ and $I = 1$, since $31/7 < 5 < 11/2$, such that x_1 replaces s_4 in the basis. For the sake of clarity, pivots are directly performed in the tableau. A set of elementary row operations must be performed in order to form a new permutation matrix (whose columns are the same as that of the identity matrix) with columns corresponding to basic variables and update the reduced costs. These operations are summarised as i) $r_0 \leftarrow r_0 + \frac{31}{7} r_4$ ii) $r_1 \leftarrow r_1 + \frac{3}{7} r_4$ iii) $r_2 \leftarrow r_2 + \frac{6}{7} r_4$ iv) $r_3 \leftarrow r_3 - \frac{2}{7} r_4$ v) $r_4 \leftarrow -\frac{1}{7} r_4$, such that the following tableau is obtained

x_1	x_2	x_3	s_1	s_2	s_3	s_4	b
0	0	0	0	4/7	15/7	31/7	
0	0	1	0	-3/7	1/7	3/7	-10/7
0	1	0	0	1/7	2/7	6/7	-20/7
0	0	0	1	2/7	11/7	-2/7	58/7
1	0	0	0	1/7	2/7	-1/7	15/7

It is clear from the reduced costs that this new basis is still dual feasible. Nevertheless, inspecting the last column of the tableau reveals that the basis is also primal infeasible. Indeed, two of the basic variables have negative values. Moreover, the coefficients of non-basic variables in the second row are all positive. This implies that the dual problem is in fact unbounded. As a result, the primal problem is infeasible. The geometric interpretation of this result is depicted in Figure 14.2c for a simplified problem.

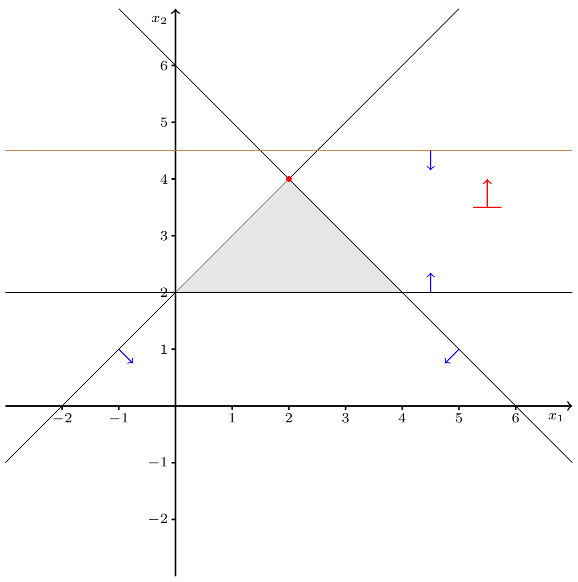
Solution to Problem 2

1. It is clear from the first (data) column of the second table that 2 batches of C-type processors and 5 batches of B-type processors are manufactured. Bearing this in mind, and using the objective coefficients stored in the third column of the second table, the profit resulting from the sale of these processors can be readily computed as $f^\star = 102C^\star + 89B^\star = 102 \times 2 + 89 \times 5 = 649$.

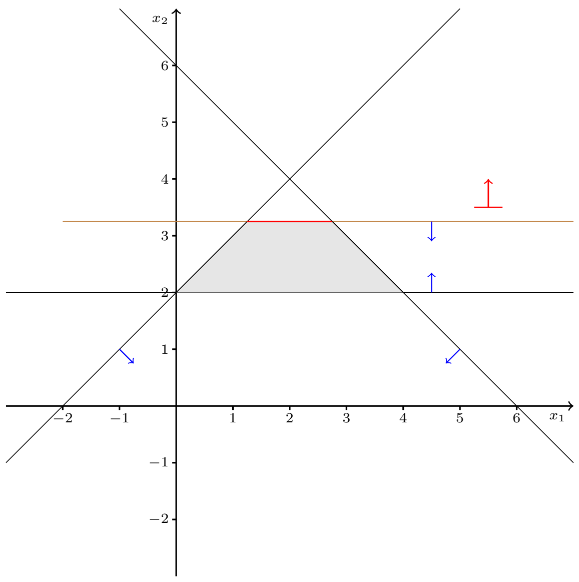
2. In order to devise an economic interpretation of dual variables, it is worth recalling some of the properties of linear programs. For the sake of conciseness, the linear program at hand is first expressed in a more abstract form. Let $c \in \mathbb{R}^n$ denote the primal cost vector, let $b \in \mathbb{R}^m$ denote the vector of right-hand side coefficients and let $x \in \mathbb{R}^n$ be the set of primal variables. The primal problem then writes as

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & x \geq 0. \end{aligned}$$

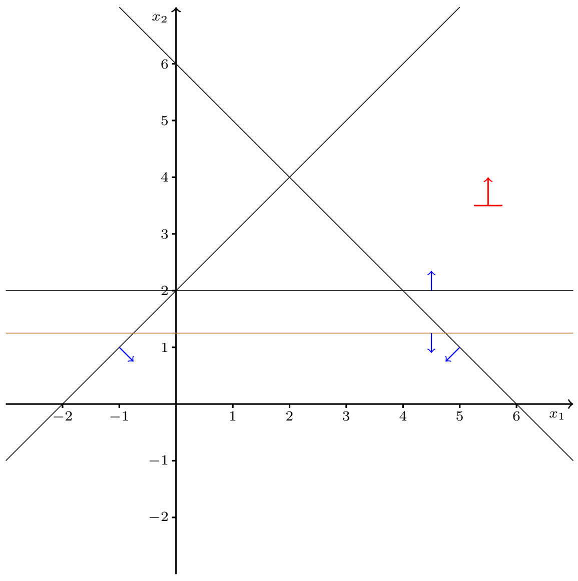
Note that an equality constraint can be expressed via two inequality constraints, hence the revenue maximization problem can always be formulated in this way. Let $p \in \mathbb{R}^m$ denote the set of dual variables,



(a) Unique optimum and redundant constraint



(b) Multiple optima (dual degenerate solution)



(c) Infeasible Problem

Figure 14.2: Impact of constraint addition on problem solution and feasibility.

such that the resulting dual problem is

$$\begin{array}{rl} \min & p^T b \\ \text{s.t.} & A^T p \geq c \\ & p \geq 0. \end{array}$$

In addition, let $x^\star$ and $p^\star$ be optimal solutions to the primal and dual problems, respectively. The strong duality property of linear programs implies that

$$c^T x^\star = (p^\star)^T b,$$

that is, the primal and dual objectives have the same value at optimality. Now, let $b'_i = b + \Delta b_i e_i,\ i = 1, \dots, m$, be an updated right-hand side coefficients vector such that the (primal) optimal basis remains unchanged, with e_i a vector whose i^{th} entry is 1 and all others are 0. Replacing b by b'_i in the primal problem will lead to a new primal solution $x^\star_i$. Note that updating the right-hand side vector has no impact on the dual solution, since b'_i only appears in the dual objective and the primal basis is unchanged. Hence, re-writing the strong duality property for the updated primal problem leads to

$$c^T x_i^\star = (p^\star)^T b'_i = (p^\star)^T b + \Delta b_i p_i^\star = c^T x^\star + \Delta b_i p_i^\star.$$

Note that in this case, b'_i and $x_i^\star$ are vectors while $p_i^\star$ denotes the i^{th} entry of the optimal dual variable vector $p^\star$. Now, let $\Delta c_i = c^T x^\star - c^T x_i^\star$ be the change in primal objective value resulting from the update of b. From the previous developments, it is clear that

$$p_i^\star = \frac{\Delta c_i}{\Delta b_i}.$$

In other words, provided that the optimal basis remains the same, a dual variable quantifies the extent to which the primal objective can be expected to change as the right-hand side coefficient of the associated constraint is updated. In the case at hand, the primal objective has a clear economic interpretation, as it represents the profits that are expected from the sale of different processor types. Thus, the dual variable of a given constraint can be interpreted as the marginal amount of money which may be gained or lost as the right-hand side coefficient is updated, e.g. as more silicon is made available (in $\$/kg$) or as the number of hours in the etching room is decreased (in $\$/hr$).

3. In order to evaluate whether it is worth investing in 20 additional kilograms of silicon, it suffices to compute the difference between the potential gain and the expenses resulting from the purchase of additional silicon. The latter can be readily obtained as $20 \times 1.1 = \$22$. Following the developments in **2.**, the potential gain can be readily assessed from the third table. Indeed, at optimality, the dual variable of the constraint expressing the silicon budget has value $p_1^\star = 1.429$, while it is proposed to increase the budget by 20 kilograms, i.e., $\Delta b_1 = 20$. The fourth column indicates that the maximum increase in silicon budget that will leave the basis unchanged is 23.33 kilograms, and since $\Delta b_1 < 23.33$, the change in primal objective value can be obtained via $\Delta c_1 = p_1^\star \Delta b_1 = 1.429 \times 20 \approx 28.5\$$. Thus, the potential gain is greater than the cost of investing in silicon, and it is worth doing so.

4. Reducing the number of hours in the doping room corresponds to decreasing the right-hand side coefficient of the fourth constraint by 4, i.e. $\Delta b_4 = -4$. As can be seen from the third table, $\Delta b_4 < -3.5$, and this decrease in coefficient value leads to a different optimal basis. Hence, the reasoning developed in **2.** and **3.** no longer applies. To gain a better understanding of the situation, let $g : \mathbb{R} \mapsto \mathbb{R} \cup \{-\infty\}$ be a function such that

$$\begin{array}{rcc} g(\Delta b_4) = & \max & c^T x \\ & \text{s.t.} & Ax \leq b + \Delta b_4 e_4 \\ & & x \geq 0. \end{array}$$

By convention, $g(\Delta b_4) = -\infty$ if the optimization problem is infeasible. In other words, for any perturbation Δb_4 of the right-hand side coefficient b_4 of the doping constraint, g returns the corresponding revenue. It can be shown that g is in fact a piecewise affine, concave function, as depicted (schematically) in Figure 14.3, in which g is shown in blue. In Figure 14.3, each of the affine segments corresponds to an optimal basis. Let us assume that the purple segment corresponds to the optimal basis corresponding to the data shown in the tables. In particular, it is worth noticing that $g(0) = 649$ and $g(-23) = 0$. Hence, if the right-hand side coefficient of the doping constraint is changed by an amount b_4 such that $-3.5 \leq \Delta b_4 \leq 5.6$, the basis remains unchanged. However, if $\Delta b_4 < -3.5$, the segment corresponding to the new optimal basis is the one to the left of the purple segment. From a geometric standpoint, following the approach of **2.** and **3.** consists in staying on the line supporting the purple segment even though the basis has changed. Hence, doing so can at best provide an upper bound on the actual objective change. Indeed, $g(0) + p_4^\star \Delta b_4 = 649 - 20.143 \times 4 = 568.428 > 563.6$, where the latter value was obtained by solving the optimization problem numerically for $\Delta b_4 = -4$.

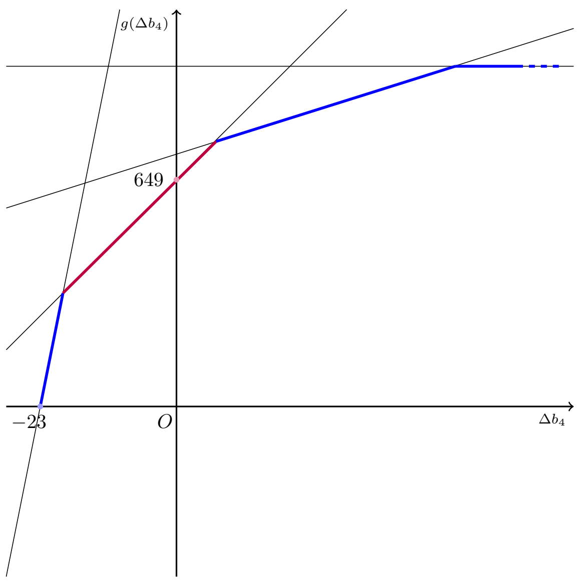


Figure 14.3: Schematic representation of changes in revenue as a result of changes in right-hand side coefficient of doping constraint. The problem becomes infeasible when $\Delta b_4 < -23$, and the revenue cannot increase any further beyond a certain value of Δb_4, as other constraints become active.

Chapter 15

Convex Models

Solution to Problem 1

a. Recall that norms have three key properties, namely

1. Triangle Inequality (subadditivity): $g(x+y) \leq g(x) + g(y)$, $\forall x, y \in \mathbb{R}^n$.
2. Absolute Homogeneity: $g(\alpha x) = |\alpha| g(x)$, $\forall \alpha \in \mathbb{R},\ x \in \mathbb{R}^n$.
3. Positivity: $g(x) \geq 0,\ \forall x \in \mathbb{R}^n$, and $g(x) = 0 \Rightarrow x = 0$.

In addition, recall that a function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is convex if $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$, $\forall x, y \in \mathbb{R}^n,\ \forall \lambda \in [0,1]$. Successively invoking the subadditivity and absolute homogeneity of g yields

$$g(\lambda x+(1-\lambda)y) \leq g(\lambda x)+g((1-\lambda)y) = |\lambda| g(x)+|1-\lambda| g(y) = \lambda g(x)+(1-\lambda) g(y),\ \forall x, y \in \mathbb{R}^n,\ \forall \lambda \in [0,1].$$

In other words, any function g having the properties of a norm is convex. In particular, the 1-norm, $||x||_1 = |x_1| + \ldots + |x_n|$, and the 2-norm, $||x||_2 = \sqrt{x_1^2 + \ldots + x_n^2}$, with x_i, $i = 1, \ldots n$, the entries of x, are convex. It is also worth noting that the 1-norm reduces to a simple absolute value function in 1D.

b. Let $C_1, C_2 \subset \mathbb{R}^n$ be two convex sets and let $C = C_1 \cap C_2$. Recall that C is convex if $\lambda y + (1-\lambda) z \in C$, $\forall y, z \in C$ and $\forall \lambda \in [0,1]$. Note that $C = \emptyset$ is trivially convex. Now, let $y, z \in C \neq \emptyset$. By definition of C, $y, z \in C_1$ and $y, z \in C_2$. Thus, since C_1 and C_2 are convex, $\lambda y + (1-\lambda) z \in C_1$ and $\lambda y + (1-\lambda) z \in C_2$, $\forall \lambda \in [0,1]$, respectively. Hence, $\lambda y + (1-\lambda) z \in C$, $\forall \lambda \in [0,1]$, and C is convex. This result can be easily extended to the intersection of $N \in \mathbb{N}$ convex sets.

c. Let $K_1, K_2 \subset \mathbb{R}^n$ be two cones such that $K = K_1 \cap K_2 \neq \emptyset$. Recall that K is a cone if $\alpha x \in K$, $\forall x \in K$, $\forall \alpha \in \mathbb{R}_+$. Let $x \in K$. By definition of K, $x \in K_1$ and $x \in K_2$. In addition, since K_1 is a cone, $\alpha x \in K_1$, $\forall \alpha \in \mathbb{R}_+$. Likewise, $\alpha x \in K_2$, $\forall \alpha \in \mathbb{R}_+$. Hence, $\alpha x \in K$, $\forall \alpha \in \mathbb{R}_+$, and K is a cone. In particular, if K_1 and K_2 are convex cones, K is a convex cone as well. It is also worth noticing that if K is a singleton, then $K = \{0\}$. This result extends trivially to the intersection of $N \in \mathbb{N}$ cones.

d. The basic idea is to show that $S_h \subseteq S_g$ and $S_h \supseteq S_g$, which implies that $S_h = S_g$. We start by showing that $S_h \subseteq S_g$. Let $y \in S_h$, such that $h_1(y) \leq h_2(y)$. Since g is strictly increasing, it follows that $g(h_1(y)) \leq g(h_2(y))$, that is, $y \in S_g$. We now show that $S_h \supseteq S_g$. Let $z \in S_g$, that is, $g(h_1(z)) \leq g(h_2(z))$. Since g is strictly increasing, this inequality implies that $h_1(z) \leq h_2(z)$, and $z \in S_h$. In other words, $S_h = S_g$.

e. Let $S_o = \arg\min\{f(x) \mid x \in F\} \neq \emptyset$ and $S_e = \arg\min\{t \mid (x,t) \in \text{ epi}(f),\ x \in F\} \neq \emptyset$. We first show that $x^\star \in S_o \Rightarrow (x^\star, f(x^\star)) \in S_e$. Let us assume that $x^\star \in S_o$ and $(x^\star, f(x^\star)) \notin S_e$. This implies that $\exists (y,s) \in \text{ epi}(f)$ with $y \in F$, such that $s < f(x^\star)$. In particular, the smallest possible value of s is $f(y)$, which is equivalent to stating that $\exists y \in F,\ f(y) < f(x^\star)$. This leads to a contradiction, as it implies that $x^\star \notin S_o$. We next show that $(x^\star, f(x^\star)) \in S_e \Rightarrow x^\star \in S_o$. Let us assume that $(x^\star, f(x^\star)) \in S_e$ and $x^\star \notin S_o$. It follows that $\exists z \in F$ such that $f(z) < f(x^\star)$. This again leads to a contradiction, as it implies that $(x^\star, f(x^\star)) \notin S_e$. Hence, $x^\star \in S_o \Leftrightarrow (x^\star, f(x^\star)) \in S_e$. For maximisation problems, the counterpart of the epigraph is the *hypograph*, $\text{hypo}(f) = \{(x,t) \in \mathbb{R}^{n+1} | \ t \leq f(x)\}$. A similar result holds, namely that $x^\star \in \arg\max\{f(x) \mid x \in F\} \Leftrightarrow (x^\star, f(x^\star)) \in \arg\max\{t \mid (x,t) \in \text{ hypo}(f),\ x \in F\}$. The result above was obtained without assuming that f is convex. However, it is worth noting that a function f is convex if and only if its epigraph is convex.

f. We start by showing that $S_h \subseteq S_g$. Let $x^\star \in S_h$, such that $h(x^\star) \geq h(x),\ \forall x \in F$. Since g is strictly increasing, $g(h(x^\star)) \geq g(h(x)),\ \forall x \in F$, that is, $x^\star \in S_g$. We now show that $S_h \supseteq S_g$. Let $x^\star \in S_g$, such that $g(h(x^\star)) \geq g(h(x)),\ \forall x \in F$. Again, since g is strictly increasing, this inequality implies that $h(x^\star) \geq h(x),\ \forall x \in F$. In other words, $x^\star \in S_h$ and $S_h = S_g$.

g. We start by showing that $S_f \subseteq S_h$. Let $x \in S_f$, such that $f(h(x)) \geq f(h(y)),\ \forall y \in F$. Since f is strictly decreasing, it follows that $h(x) \leq h(y),\ \forall y \in F$, hence $x \in S_h$. We now show that $S_f \supseteq S_h$. Let $z \in S_h$, such that $h(z) \leq h(y),\ \forall y \in F$. Since f is strictly decreasing, the following inequality holds $f(h(z)) \geq f(h(y))$, thus $z \in S_f$ and $S_f = S_h$.

Problem 2

Show that the following optimisation problems are convex (or not). Where possible, propose a convex reformulation with the same optimal solution set.

$$\text{(a)} \quad \begin{aligned} \min\ & |x_1 - 3x_2| \\ \text{s.t.}\ & x_1 + 2x_2 \leq 3 \\ & -2x_1 + 3x_2 \leq 0 \\ & x_1 \geq 2, x_2 \geq 0 \end{aligned}$$

$$\text{(b)} \quad \begin{aligned} \max\ & |x_1 + 2x_2| \\ \text{s.t.}\ & Ax = b \\ & x \in \mathbb{R}^n_+ \end{aligned}$$

$$\text{(c)} \quad \begin{aligned} \max\ & |x_1 - 2x_2| \\ \text{s.t.}\ & Ax = b \\ & x \in \mathbb{R}^n_+ \end{aligned}$$

$$\text{(d)} \quad \begin{aligned} \min\ & \sqrt{x_1^2 + x_2^2} \\ \text{s.t.}\ & x_1 + x_2 = \frac{1}{2} \end{aligned}$$

$$\text{(e)} \quad \begin{aligned} \min\ & x_1 + 2x_2 \\ \text{s.t.}\ & x_1^2 + 9x_2^2 + x_3^2 \leq 4 \end{aligned}$$

$$\text{(f)} \quad \begin{aligned} \min\ & 2x_1 - x_2 \\ \text{s.t.}\ & (x_1 - 3x_2)^3 \leq (2x_1 + x_2)^3 \\ & x \in \mathbb{R}^2 \end{aligned}$$

$$\text{(g)} \quad \begin{aligned} \max\ & x_1 x_2 \\ \text{s.t.}\ & Ax = b \\ & x \in \mathbb{R}^n_+ \end{aligned}$$

$$\text{(h)} \quad \begin{aligned} \min\ & 1 \\ \text{s.t.}\ & \frac{||Ax+b||^2}{c^Tx+d} \leq t \\ & c^Tx + d > 0 \end{aligned}$$

$$\text{(i)} \quad \begin{aligned} \max\ & \frac{1}{c^Tx} \\ \text{s.t.}\ & Ax \leq b \\ & x \in \mathbb{R}^n \end{aligned}$$

$$\text{(j)} \quad \begin{aligned} \min\ & \log(c^Tx) \\ \text{s.t.}\ & Ax \leq b \\ & x \in \mathbb{R}^n_+ \end{aligned}$$

$$\text{(k)} \quad \begin{aligned} \min\ & \lambda_{\max}(X) \\ & AX = B \\ & X = X^T \\ & X \in \mathbb{R}^{n\times n} \end{aligned}$$

$$\text{(l)} \quad \begin{aligned} \min\ & \lambda_{\max}(X) + x_{11}^2 \\ & AX = B \\ & X = X^T \\ & X \in \mathbb{R}^{n\times n} \end{aligned}$$

$$\text{(m)} \quad \begin{aligned} \min\ & \max\{\lambda_{\max}(X), x_{11}^2\} \\ & AX = B \\ & X = X^T \\ & X \in \mathbb{R}^{n\times n} \end{aligned}$$

Solution to Problem 2

a. Notice that the objective function is convex, and the feasible set is convex, and recall that minimising a convex function over a convex set is a convex problem. However, the objective function is nonlinear and non-differentiable at (0, 0). An equivalent formulation with more desirable properties is therefore sought. Casting the problem in epigraph form yields

$$\begin{aligned} \min\ & t \\ \text{s.t.}\ & |x_1 - 3x_2| \leq t \\ & x_1 + 2x_2 \leq 3 \\ & -2x_1 + 3x_2 \leq 0 \\ & x_1 \geq 2, x_2 \geq 0. \end{aligned}$$

If $x_1 - 3x_2 > 0$, the first inequality becomes $x_1 - 3x_2 \leq t$. By contrast, if $x_1 - 3x_2 < 0$, the first inequality becomes $-x_1 + 3x_2 \leq t$, or $-t \leq x_1 - 3x_2$. Hence, the first inequality, whose left-hand side is nonlinear, can be replaced by two linear inequalities. The resulting linear programming problem therefore writes as

$$\begin{aligned} \min\ & t \\ \text{s.t.}\ & x_1 - 3x_2 \leq t \\ & -t \leq x_1 - 3x_2 \\ & x_1 + 2x_2 \leq 3 \\ & -2x_1 + 3x_2 \leq 0 \\ & x_1 \geq 2, x_2 \geq 0. \end{aligned}$$

b. Notice that $|x_1 + 2x_2| = x_1 + 2x_2$ for $x_1, x_2 \geq 0$. In other words, the problem is equivalent to the following linear program

$$\begin{aligned} \max\ & x_1 + 2x_2 \\ \text{s.t.}\ & Ax = b \\ & x \in \mathbb{R}^n_+. \end{aligned}$$

c. The objective function of the problem at hand is convex. However, in contrast to **a.**, the present problem is a maximization problem. Put in hypograph form, the problem reads

$$\begin{aligned} \max\ & t \\ \text{s.t.}\ & t \leq |x_1 - 2x_2| \\ & Ax = b \\ & x \in \mathbb{R}^n_+. \end{aligned}$$

The objective function is linear, but it is worth checking whether the feasible set is convex. Let $g(x_1, x_2) = |x_1 - 2x_2|$. In fact, $\text{hypo}(g)$ can be shown to be nonconvex. Indeed, for instance, $(x_1, x_2, t) = (0, -1, 2) \in \text{hypo}(g)$ and $(x_1, x_2, t) = (0, 1, 2) \in \text{hypo}(g)$. However, $0.5(0, -1, 2) + 0.5(0, 1, 2) = (0, 0, 2) \notin \text{hypo}(g)$, as $2 > 0$. Hence, the problem is nonconvex. In fact, at the notable exception of affine objective functions, maximising a convex function over a convex set will yield a nonconvex problem. By contrast, it is simple to show that maximising a concave function over a convex set is a convex optimisation problem. Indeed, the opposite of a concave function is always a convex function, and a maximisation problem can always be transformed into a minimisation problem. If $f(x)$ denotes a concave function, $\max f(x) = -\min -f(x)$, with $-f(x)$ convex. Provided that the feasible set is convex, the resulting problem will also be convex.

d. Notice that the objective function is the 2-norm in 2D and that the feasible set is a line. The problem is therefore convex. Nevertheless, the objective function is nonlinear and non-differentiable at (0, 0). An equivalent representation with better properties is thus sought. Writing the problem in epigraph form yields

$$\begin{aligned} \min\ & t \\ \text{s.t.}\ & \sqrt{x_1^2 + x_2^2} \leq t \\ & x_1 + x_2 = \frac{1}{2}, \end{aligned}$$

This is a conic program, as the objective function is linear, the first constraint defines a *quadratic cone*, while the second constraint defines an affine hyperplane. The quadratic cone is also sometimes referred to as the *ice cream cone* or the *second-order cone.* Let $\mathcal{Q}^n = \{x \in \mathbb{R}^n |\ x_1 \geq \sqrt{x_2^2 + \ldots + x_n^2}\}$ be the quadratic cone of dimension n. To emphasise the fact that the first constraint represents a quadratic cone, the problem may be re-written as

$$\begin{aligned} \min\ & t \\ \text{s.t.}\ & (t, x_1, x_2) \in \mathcal{Q}^3 \\ & x_1 + x_2 = \frac{1}{2}. \end{aligned}$$

e. This minimisation problem is convex, as both the objective function and the feasible set are convex. Indeed, notice that the left-hand side of the inequality constraint can be understood as a modified 2-norm, such that the feasible set is a so-called *norm ball.* Its convexity can be directly shown using the arguments developed in **1.a**. With this in mind, it is clear that the modified norm ball can be expressed as the intersection of an affine hyperplane and a quadratic cone. Indeed, introducing additional variables $y, z \in \mathbb{R}$ allows us to express the optimisation problem at hand as a conic program

$$\begin{aligned} \min\ & x_1 + 2x_2 \\ \text{s.t. } & x_1^2 + y^2 + x_3^2 \le z^2 \\ & y = 3x_2 \\ & z = 2. \end{aligned}$$

Then, by virtue of **1.d**, the problem may be equivalently written as

$$\begin{aligned} \min\ & x_1 + 2x_2 \\ \text{s.t. } & (z, x_1, y, x_3) \in \mathcal{Q}^4 \\ & y = 3x_2 \\ & z = 2. \end{aligned}$$

f. Let $g : \mathbb{R} \mapsto \mathbb{R}$ be such that $g(x) = x^3$. Clearly, g is a strictly increasing function of its argument. Hence, by virtue of **1.d**, the inequality $g(x_1 - 3x_2) \le g(2x_1 + x_2)$ can be transformed into $x_1 - 3x_2 \le 2x_1 + x_2$ without altering the feasible set. In other words, the original problem can be cast as a linear program

$$\begin{aligned} \min\ & 2x_1 - x_2 \\ \text{s.t. } & x_1 - 3x_2 \le 2x_1 + x_2 \\ & x \in \mathbb{R}^2. \end{aligned}$$

g. We start by rewriting the optimisation problem in a more general form. Let $Q \in \mathbb{R}^{n\times n}$, $Q = Q^T$, and let $g : \mathbb{R}^n \mapsto \mathbb{R}$ be the quadratic form $q(x) = x^T Q x$. The problem at hand is a particular instance of quadratic programs of the form

$$\begin{aligned} \max\ & x^T Q x \\ \text{s.t. } & Ax = b \\ & x \in \mathbb{R}^n_+. \end{aligned}$$

In the particular case of negative semidefinite matrices, that is, if $Q \preceq 0$, the problem is concave, as Q is the Hessian of the objective function. By contrast, for indefinite matrices, q is indefinite and neither concave nor convex over its full domain. In the case at hand,

$$Q = \begin{pmatrix} 0 & 1/2 & 0_{n-2}^T \\ 1/2 & 0 & 0 \\ 0_{n-2} & 0 & 0_{(n-2)\times(n-2)} \end{pmatrix},$$

which is indefinite. Indeed, it is simple to check that it has only two nonzero eigenvalues, which are of opposite sign. We therefore seek a formulation with the same solution set and better properties. Notice that, in this particular case, i) $q(x)$ reduces to the bilinear function $x_1 x_2$ ii) $q(x) \ge 0$, $\forall x \in \mathbb{R}^n_+$. In fact, $x_1 x_2$ bears some resemblance to the geometric mean $g(x) = (\Pi_{i=1}^n x_i)^{1/n}$, with x_i the entries of $x \in \mathbb{R}^n_+$, which is known to be concave and reduces to $\sqrt{x_1 x_2}$ in 2D. More precisely, forming the composition of q and the square root function, which is strictly increasing, yields g. Hence, following the arguments of **1.f**, maximising $x_1 x_2$ and $\sqrt{x_1 x_2}$ is in fact equivalent. Thus, the convex problem

$$\begin{aligned} \max\ & \sqrt{x_1 x_2} \\ \text{s.t. } & Ax = b \\ & x \in \mathbb{R}^n_+ \end{aligned}$$

has the same solution set as the original problem. Casting it in hypograph form yields

$$\begin{aligned} \max\ & t \\ \text{s.t. } & t \leq \sqrt{x_1 x_2} \\ & Ax = b \\ & x \in \mathbb{R}^n_+. \end{aligned}$$

By virtue of **1.d**, taking the square of the left-hand and right-hand sides of the first inequality produces an equivalent problem, which reads

$$\begin{aligned} \max\ & t \\ \text{s.t. } & t^2 \leq x_1 x_2 \\ & Ax = b \\ & x \in \mathbb{R}^n_+. \end{aligned}$$

Introducing the change of variables $x_1 = y + z$ and $x_2 = y - z$ allows us to transform the first inequality into a constraint defining a quadratic cone. Substituting the expressions of x_1 and x_2 into the right-hand side of the first inequality constraint yields $x_1 x_2 = y^2 - z^2$, such that the problem at hand becomes

$$\begin{aligned} \max\ & t \\ \text{s.t. } & t^2 + z^2 \leq y^2 \\ & x_1 = y + z \\ & x_2 = y - z \\ & Ax = b \\ & x \in \mathbb{R}^n_+. \end{aligned}$$

The objective function is linear, the first constraint defines a quadratic cone, while all other constraints define affine hyperplanes. Hence, this is a conic program. Note that the optimal objective value of this equivalent problem must be squared to obtain the optimal objective value of the original problem. Alternatively, we could have introduced variables $u = x_1 + x_2$ and $v = x_1 - x_2$ such that $u^2 - v^2 = x_1 x_2$, leading to the conic program

$$\begin{aligned} \max\ & t \\ \text{s.t. } & t^2 + v^2 \leq u^2 \\ & u = x_1 + x_2 \\ & v = x_1 - x_2 \\ & Ax = b \\ & x \in \mathbb{R}^n_+. \end{aligned}$$

h. Notice that the objective function of the problem at hand is constant. Such problems are called *feasibility problems*, since solving them essentially amounts to finding a feasible solution. Now, let $u = \frac{1}{2}(c^T x + d) > 0$ and $v = Ax + b$. The problem becomes

$$\begin{aligned} \min\ & 1 \\ \text{s.t. } & \frac{||v||_2^2}{2u} \leq t \\ & v = Ax + b \\ & u = \frac{1}{2}(c^T x + d) \\ & u > 0. \end{aligned}$$

Multiplying both sides of the first inequality by $2u$ then yields,

$$\begin{aligned} \min\ & 1 \\ \text{s.t.}\ & ||v||_2^2 \leq 2ut \\ & v = Ax + b \\ & u = \frac{1}{2}(c^T x + d) \\ & u > 0. \end{aligned}$$

where the first inequality now defines a *rotated quadratic cone*. A rotated quadratic cone can be transformed into a quadratic cone by introducing the change of variables $u = \frac{1}{\sqrt{2}}(y+z)$ and $t = \frac{1}{\sqrt{2}}(y-z)$. This change of variables can in fact be interpreted as an orthogonal transformation of the coordinate system expressing a rotation of $\pi/4$ in the (u,t) plane, and leads to

$$\begin{aligned} \min\ & 1 \\ \text{s.t.}\ & ||v||_2^2 + z^2 \leq y^2 \\ & v = Ax + b \\ & u = \frac{1}{2}(c^T x + d) \\ & u = \frac{1}{\sqrt{2}}(y+z) \\ & t = \frac{1}{\sqrt{2}}(y-z) \\ & u > 0. \end{aligned}$$

In any case, both formulations of the problem above are conic programs. As in **2.g**, an alternative change of variables could have been introduced to obtain an equivalent conic program.

i. Let $f : \mathbb{R} \mapsto \mathbb{R} \cup \{-\infty, +\infty\}$ be a mapping such that $f(z) = 1/z$, and let $h : \mathbb{R}^n \mapsto \mathbb{R}$ be a mapping such that $h(x) = c^T x$ for some parameter $c \in \mathbb{R}^n$. The objective function can be obtained as the composition of f and h. Considering the discontinuity at 0 and the sign of f on each side of it, i.e., $\lim_{z\to 0^-} f(z) = -\infty$ and $\lim_{z \to 0^+} f(z) = +\infty$, it is clear that the function is nonconvex. As a result, the problem is nonconvex. However, the restriction of f to either side of 0 has nicer properties, which suggests trying to decompose the original problem into two simpler problems. Note that if $\exists x$ such that $Ax \leq b$ and $c^T x = 0$, the original problem is ill-posed. In what follows, the original problem is assumed well-posed, such that it can be readily decomposed into

$$\begin{aligned} \max\ & \frac{1}{c^T x} \\ \text{s.t.}\ & Ax \leq b \\ & c^T x \geq 0 \\ & x \in \mathbb{R}^n_+ \end{aligned} \qquad \text{and} \qquad \begin{aligned} \max\ & \frac{1}{c^T x} \\ \text{s.t.}\ & Ax \leq b \\ & c^T x \leq 0 \\ & x \in \mathbb{R}^n_+. \end{aligned}$$

In each subproblem, the objective function is the composition of a strictly decreasing function and an affine one. Thus, by virtue of **1.g**, the two subproblems can be transformed into

$$\begin{aligned} \min\ & c^T x \\ \text{s.t.}\ & Ax \leq b \\ & c^T x \geq 0 \\ & x \in \mathbb{R}^n_+ \end{aligned} \qquad \text{and} \qquad \begin{aligned} \min\ & c^T x \\ \text{s.t.}\ & Ax \leq b \\ & c^T x \leq 0 \\ & x \in \mathbb{R}^n_+, \end{aligned}$$

which are linear programs. The best solution of these two programs can then be retained as the solution to the original problem.

j. Note that the domain of the log function is $\mathbb{R}_+ \setminus \{0\}$. Thus, in this case, it is only defined if $c^T x > 0$ in the polyhedron $Ax \leq b$. In the following, it is assumed that this condition is satisfied. The log function is obviously nonconvex. Indeed, $\log(0.5\times1+0.5\times2) = \log(1.5) \approx 0.585 > 0.5\log(1)+0.5\log(2) = 0.5$, and the problem is nonconvex. However, the log function is strictly increasing over its domain. Hence, invoking the same arguments as the ones developed in **2.g** suffices to show that $\arg\min\{c^T x \mid Ax \leq b,\ x \in \mathbb{R}^n_+\}$ and $\arg\min\{\log(c^T x) \mid Ax \leq b,\ x \in \mathbb{R}^n_+\}$ are the same. In other words, a convex optimisation problem with the same solution set as the original problem can be readily formulated. This equivalent problem is the linear program

$$\begin{aligned} \min\ & c^T x \\ \text{s.t.}\ & Ax \leq b \\ & x \in \mathbb{R}^n_+. \end{aligned}$$

It is worth noting that if the solution to this problem is nonpositive, the original problem is ill-posed.

k. Let $F = \{X \in \mathbb{R}^{n\times n} \mid AX = B,\ X = X^T\} \neq \emptyset$ be the feasible set of the original problem, which is assumed non-empty. The matrix equation $AX = B$ is equivalent to a collection of systems of linear equations. From a geometric perspective, the solution to each system is an affine subspace, such that the matrix equation describes the intersection of a collection of affine subspaces. Now, recall that the eigenvalues of a square matrix $X \in \mathbb{R}^{n\times n}$ are the roots of the characteristic polynomial, i.e., they are obtained by solving $\det(X - \lambda I) = 0$, with det the determinant operator and I the $n \times n$ identity matrix. Hence, $\lambda_{max}(X) = \arg\max\{\lambda \in \mathbb{R} \mid \det(X - \lambda I) = 0\}$. The objective function of the problem at hand is particularly cumbersome. However, the problem can be reformulated in order to obtain a more tractable model. Indeed, writing it in epigraph form yields

$$\begin{aligned} \min\ & t \\ \text{s.t.}\ & \lambda_{\max}(X) \leq t \\ & AX = B \\ & X = X^T \\ & X \in \mathbb{R}^{n\times n}. \end{aligned}$$

The inequality constraint can be readily rewritten as $\lambda_{\max}(X) - t \leq 0$. For a given X, let $\lambda_M = \lambda_{max}(X)$. Notice that $\lambda_M - t$ is an eigenvalue of $X - tI$. Indeed, let v_M denote the eigenvector of X corresponding to λ_M. Then, $(X - tI)\mathrm{v}_M = X\mathrm{v}_M - tI\mathrm{v}_M = (\lambda_M - t)\mathrm{v}_M$. All other eigenvalues of $X - tI$ can be obtained in similar fashion. Now, let us assume that these eigenvalues have been ordered and let I denote the set indexing them. By definition of λ_M, $\lambda_M - t \geq \lambda_i - t$, $\forall i \in I$ and $\forall t \in \mathbb{R}$. In other words, $\lambda_M - t = \lambda_{max}(X - tI)$. The inequality constraint therefore becomes $\lambda_{max}(X - tI) \leq 0$, which implies that the largest eigenvalue of the $X - tI$ matrix must be nonpositive. This condition is equivalent to stating that $X - tI$ is a negative semidefinite matrix, that is, $X - tI \preceq 0$. Moreover, it can be shown that this constraint defines a cone. Indeed, an alternative way of expressing the negative semidefinitess of $X - tI$ consists in stating that $z^T(X - tI)z \leq 0, \forall z \in \mathbb{R}^n$. It follows that $\alpha z^T(X - tI)z \leq 0, \forall z \in \mathbb{R}^n$, $\forall \alpha \geq 0$. The optimisation problem therefore reduces to the conic program

$$\begin{aligned} \min\ & t \\ \text{s.t.}\ & -X + tI \succeq 0 \\ & AX = B \\ & X = X^T \\ & X \in \mathbb{R}^{n\times n}. \end{aligned}$$

l. A variant of the epigraph trick can be applied to each term in the objective function, yielding

$$\begin{aligned} \min\ & t+s \\ \text{s.t.}\ & \lambda_{max}(X) \leq t \\ & x_{11}^2 \leq s \\ & AX = B \\ & X = X^T \\ & X \in \mathbb{R}^{n\times n}. \end{aligned}$$

Building upon the developments of **2.k**, the problem becomes

$$\begin{aligned} \min\ & t+s \\ \text{s.t.}\ & -X + tI \succeq 0 \\ & x_{11}^2 \leq s \\ & AX = B \\ & X = X^T \\ & X \in \mathbb{R}^{n\times n}, \end{aligned}$$

which is a conic program.

m. Minimising the maximum of two functions is equivalent to minimising a variable bounding both of them. Hence, the problem can be rewritten as

$$\begin{aligned} \min\ & t \\ \text{s.t.}\ & \lambda_{max}(X) \leq t \\ & x_{11}^2 \leq t \\ & AX = B \\ & X = X^T \\ & X \in \mathbb{R}^{n\times n}, \end{aligned}$$

or, equivalently,

$$\begin{aligned} \min\ & t \\ \text{s.t.}\ & -X + tI \succeq 0 \\ & x_{11}^2 \leq t \\ & AX = B \\ & X = X^T \\ & X \in \mathbb{R}^{n\times n}, \end{aligned}$$

which is again a conic program.

Solution to Problem 3

Let $t \in \mathbb{R}_+$ be a new optimisation variable, and let $f : \mathbb{R}^n \mapsto \mathbb{R}_+$ denote the quadratic form $f(x) = x^T Qx$. Recall that the epigraph of f is the set $\text{epi}(f) = \{(x,t) \in \mathbb{R}^{n+1} | f(x) \leq t\}$. With this in mind, the optimisation problem can be equivalently written as

$$\begin{aligned} \min\quad & t \\ \text{s.t.}\quad & x^T Qx \leq t \\ & ||x||_2 \leq \gamma. \end{aligned}$$

Then, since $Q \succ 0$, its Cholesky factorization is unique and can be expressed as $Q = LL^T$, with $L \in \mathbb{R}^{n\times n}$ a lower triangular matrix. The first constraint therefore becomes

$$x^T Qx \leq t \Leftrightarrow x^T LL^T x \leq t \Leftrightarrow ||L^T x||_2^2 \leq t.$$

Now, let $S_1 = \{(t, x) \in \mathbb{R}^{n+1} |\ ||L^T x||_2^2 \leq t\}$, and let $K_1 = \{(s, t, x) \in \mathbb{R}^{n+2} |\ ||L^T x||_2^2 \leq 2st\}$. It is straightforward to verify that $\forall z \in K_1$ and $\forall \alpha > 0,\ \alpha z \in K_1$, i.e., K_1 is a cone. More precisely, K_1 is a rotated quadratic cone. It is now clear that S_1 can be expressed as the intersection of a rotated quadratic cone and an affine hyperplane, that is, $S_1 = \{(s, t, x) \in \mathbb{R}^{n+2} |\ (s, t, x) \in K_1,\ s = 1/2\}$. Likewise, let $S_2 = \{x \in \mathbb{R}^n |\ ||x||_2 \leq \gamma\}$ and $K_2 = \{(u, x) \in \mathbb{R}^{n+1} |\ ||x||_2 \leq u\}$. K_2 is a quadratic cone, and S_2 can be expressed as the intersection of this cone and an affine hyperplane, i.e., $S_2 = \{(u, x) \in \mathbb{R}^{n+1} |\ (u, x) \in K_2,\ u = \sqrt{\gamma}\}$. Finally, let $H = \{(s, u) \in \mathbb{R}^2 |\ s = 1/2,\ u = \sqrt{\gamma}\}$, which represents an affine hyperplane. The optimisation problem therefore writes as

$$\begin{array}{ll} \min & t \\ \text{s.t.} & (s, t, x) \in K_1 \\ & (u, x) \in K_2 \\ & (s, u) \in H, \end{array}$$

which is a conic program, as the objective function is linear and the feasible set is formed by the intersection of two cones and an affine hyperplane.

Chapter 16

Second-Order Cone Programming

Solution to Problem 1

Let $A = (0,0)$, $B = (1,1)$ and $C = (2,1)$, respectively. The problem then reduces to that of finding a point $P = (x,y) \in \mathbb{R}^2$ such that the maximum distance between P and A, B and C is minimum. The distances between these points can be expressed as $d_A(x,y) = \sqrt{x^2+y^2}$, $d_B(x,y) = \sqrt{(x-1)^2+(y-1)^2}$ and $d_C(x,y) = \sqrt{(x-2)^2+(y-1)^2}$, respectively. The optimisation problem therefore reads

$$\begin{aligned} \min\ & \max\{d_A(x,y), d_B(x,y), d_C(x,y)\} \\ \text{s.t. } & x \in \mathbb{R},\ y \in \mathbb{R}. \end{aligned}$$

It is easy to see that minimising the max of three functions is equivalent to minimising a variable bounding all three functions simultaneously. Let $t \in \mathbb{R}$ be this variable, such that the problem can be rewritten as

$$\begin{aligned} \min\ & t \\ \text{s.t. } & d_A(x,y) \le t \\ & d_B(x,y) \le t \\ & d_C(x,y) \le t \\ & x \in \mathbb{R},\ y \in \mathbb{R},\ t \in \mathbb{R}, \end{aligned}$$

or, equivalently,

$$\begin{aligned} \min\ & t \\ \text{s.t. } & \sqrt{x^2+y^2} \le t \\ & \sqrt{(x-1)^2+(y-1)^2} \le t \\ & \sqrt{(x-2)^2+(y-1)^2} \le t \\ & x \in \mathbb{R},\ y \in \mathbb{R},\ t \in \mathbb{R}, \end{aligned}$$

which is a second-order cone program.

Solution to Problem 2

a. Since Bond is 50 metres away from the shore and the boat is also 50 metres away from the shore, the main problem is to identify the entry point of Bond into the sea. Let $z \in \mathbb{R}$ denote the lateral distance Bond should cover on land before entering the sea. In addition, let $x \in \mathbb{R}_+$ and $y \in \mathbb{R}_+$ denote the distance covered on land and in the sea, respectively. Clearly, $x = \sqrt{z^2+50^2}$ and $y = \sqrt{(100-z)^2+50^2}$. In addition, the total time to reach the boat can be computed as $x/5 + y/2.778$. Thus, the problem Bond is trying to solve is

$$\begin{aligned} \min\ & \frac{x}{5} + \frac{y}{2.778} \\ \text{s.t. } & x = \sqrt{z^2+50^2} \\ & y = \sqrt{(100-z)^2+50^2} \\ & x, y \in \mathbb{R}_+, z \in \mathbb{R}. \end{aligned}$$

Since equality constraints are nonlinear, the problem is nonconvex. A simple way to produce an equivalent convex problem is to relax the equality constraints. The problem then becomes

$$\begin{aligned} \min\ & \frac{x}{5} + \frac{y}{2.778} \\ \text{s.t.}\ & x \geq \sqrt{z^2 + 50^2} \\ & y \geq \sqrt{(100-z)^2 + 50^2} \\ & x, y \in \mathbb{R}_+, z \in \mathbb{R}. \end{aligned}$$

This updated problem is equivalent to the original problem since x and y both have positive coefficients in the objective function which is minimised. Hence, the relaxed constraints will in fact be satisfied with equality. Furthermore, each inequality constraint now defines a quadratic cone, such that the problem at hand is a second-order cone program.

b. Recall that for a generic conic program

$$\begin{aligned} \min\ & c^T w \\ \text{s.t.}\ & (A_i w - b_i) \in K^i,\ i = 1, \ldots, N \\ & w \in \mathbb{R}^n, \end{aligned}$$

with $K^i \subset \mathbb{R}^m,\ i = 1, \ldots, N$, a (convex) cone, the dual problem reads

$$\begin{aligned} \max\ & \sum_{i=1}^{N} b_i^T p_i \\ \text{s.t.}\ & c = \sum_{i=1}^{N} A_i^T p_i \\ & p_i \in K_*^i,\ i = 1, \ldots, N. \end{aligned}$$

Note that some cones are self-dual, that is, $K = K_*$. In the case at hand, we are dealing with two quadratic cones, which can be shown to be self-dual. The primal problem can now be rewritten as

$$\begin{aligned} \min\ & \frac{x}{5} + \frac{y}{2.778} \\ \text{s.t.}\ & (x, z, 50) \in \mathbb{L}^2 \\ & (y, 100 - z, 50) \in \mathbb{L}^2 \\ & x, y \in \mathbb{R}_+, z \in \mathbb{R}, \end{aligned}$$

or, equivalently,

$$\begin{aligned} \min\ & c^T w \\ \text{s.t.}\ & (A_1 w - b_1) \in \mathbb{L}^2 \\ & (A_2 w - b_2) \in \mathbb{L}^2 \\ & w \in \mathbb{R}_+^2 \times \mathbb{R}, \end{aligned}$$

with

$$w = \begin{pmatrix} x \\ y \\ z \end{pmatrix},\ c = \begin{pmatrix} 1/5 \\ 1/2.77 \\ 0 \end{pmatrix},\ A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix},\ A_2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix},\ b_1 = \begin{pmatrix} 0 \\ 0 \\ -50 \end{pmatrix} \text{ and } b_2 = \begin{pmatrix} 0 \\ -100 \\ -50 \end{pmatrix}.$$

The dual problem is immediately retrieved and reads

$$\begin{aligned} \max\ & b_1^T p_1 + b_2^T p_2 \\ \text{s.t.}\ & A_1^T p_1 + A_2^T p_2 = c \\ & p_1 \in \mathbb{L}^2,\ p_2 \in \mathbb{L}^2. \end{aligned}$$

Let $p_1 = (r_1, s_1, t_1)^T$ and $p_2 = (r_2, s_2, t_2)^T$. Then, the dual problem can be equivalently expressed as

$$\begin{aligned} \max\ & -50t_1 - 100s_2 - 50t_2 \\ \text{s.t.}\ & r_1 + s_2 = 1/5 \\ & t_1 - t_2 = 1/2.77 \\ & (r_1, s_1, t_1) \in \mathbb{L}^2,\ (r_2, s_2, t_2) \in \mathbb{L}^2. \end{aligned}$$

Solution to Problem 3

For the sake of clarity, we focus on a single chance constraint $\mathbb{P}(a_i^T x \leq b_i) \geq \eta$, bearing in mind that the developments carried out in the following can be readily applied to each one of them. For a given $x \in \mathbb{R}^n$, let $u_i = a_i^T x$. Notice that $u_i \in \mathbb{R}$ is a (scalar) random variable. More precisely, u_i is obtained as the sum of Gaussian random variables, and it is therefore Gaussian, with mean $\bar{u}_i = \bar{a}_i^T x$ and variance $\sigma = x^T \Sigma_i x$. Hence, in plain language, the chance constraint $\mathbb{P}(a_i^T x \leq b_i) \geq \eta$ expresses that the probability of u_i being smaller than b_i must be greater than or equal to the confidence level η. Recall that such probabilities can be computed via cumulative distribution functions (CDFs). In the case of Gaussian variables, tabulated probability values are typically available for variables with zero mean and unit variance. We therefore seek to rearrange the chance constraint to feature a random variable with these properties. This can be simply achieved by introducing the random variable $\frac{u_i - \bar{u}_i}{\sqrt{\sigma_i}}$ and writing

$$\mathbb{P}(u_i \leq b_i) \geq \eta \Leftrightarrow \mathbb{P}\left(\frac{u_i - \bar{u}_i}{\sqrt{\sigma_i}} \leq \frac{b_i - \bar{u}_i}{\sqrt{\sigma_i}}\right) \geq \eta.$$

Note that the covariance matrix Σ_i is positive semidefinite (and symmetric) by definition, such that $\sqrt{\sigma_i}$ is well-defined. For now, let us assume that x has been selected such that $\sqrt{\sigma_i} \neq 0$. Let $\Phi : \mathbb{R} \mapsto [0, 1]$ denote the CDF of a scalar Gaussian variable with zero mean and unit variance. Then, clearly,

$$\mathbb{P}\left(\frac{u_i - \bar{u}_i}{\sqrt{\sigma_i}} \leq \frac{b_i - \bar{u}_i}{\sqrt{\sigma_i}}\right) = \Phi\left(\frac{b_i - \bar{u}_i}{\sqrt{\sigma_i}}\right).$$

In other words, the chance constraint can be expressed as

$$\Phi\left(\frac{b_i - \bar{u}_i}{\sqrt{\sigma_i}}\right) \geq \eta.$$

Now, let $\Phi^{-1} : [0, 1] \mapsto \mathbb{R}$ be the inverse of Φ. Although no closed-form expression is available for Φ^{-1}, it is known to be strictly increasing over its domain. Thus, forming the composition of Φ^{-1} and the functions on both sides of the chance constraint yields

$$\Phi\left(\frac{b_i - \bar{u}_i}{\sqrt{\sigma_i}}\right) \geq \eta \Leftrightarrow \Phi^{-1}\left(\Phi\left(\frac{b_i - \bar{u}_i}{\sqrt{\sigma_i}}\right)\right) \geq \Phi^{-1}(\eta) \Leftrightarrow \frac{b_i - \bar{u}_i}{\sqrt{\sigma_i}} \geq \Phi^{-1}(\eta),$$

as the composition of a function with its inverse is the identity map. Multiplying both sides of the inequality by $\sqrt{\sigma_i}$ then leads to

$$b_i - \bar{u}_i \geq \Phi^{-1}(\eta)\sqrt{\sigma_i}.$$

The right-hand side of this inequality is an affine function of x, as $b_i - \bar{u}_i = b_i - \bar{a}_i^T x$. Inspecting the left-hand side of the inequality reveals that it can be expressed in terms of the 2-norm of some linear function of x. Indeed,

$$\sqrt{\sigma_i} = \sqrt{x^T \Sigma_i x} = \sqrt{x^T \Sigma_i^{\frac{1}{2}} \Sigma_i^{\frac{1}{2}} x} = ||\Sigma_i^{\frac{1}{2}} x||_2.$$

Note that the positive square root $\Sigma_i^{\frac{1}{2}}$ of the covariance matrix is well-defined and symmetric because the latter is positive semidefinite and symmetric. Hence, the original problem can be recast as

$$\begin{aligned} \min\ & c^T x \\ \text{s.t.}\ & b_i - \bar{a}_i^T x \geq \Phi^{-1}(\eta)||\Sigma_i^{\frac{1}{2}} x||_2,\ i = 1, \ldots, N \\ & x \in \mathbb{R}^n. \end{aligned}$$

In the case at hand, $\eta \geq 0.5$, such that $\Phi^{-1}(\eta) \geq 0$, and each chance constraint defines a second-order cone. Thus, the original chance constrained linear program has been recast as a second-order cone program. Note that this formulation also works for $x \in \mathbb{R}^n$ such that $\sqrt{\sigma_i} = ||\Sigma_i^{\frac{1}{2}} x||_2 = 0$.

Solution to Problem 4

a. Ideally, we would like $f(x) \geq 1$, $\forall x \in X$, which is equivalent to stating that $1 - f(x) = 1 - w^T x + b \leq 0$, $\forall x \in X$. If this condition is satisfied for a given x, no penalty should be incurred. By contrast, $1 - w^T x + b \geq 0$ implies that $f(x_i) \leq 1$ and the separation constraint is violated. Thus, such outcomes should be penalised. Using the max function, $\forall x \in X$, a simple penalty can be expressed via $l_X(x) = \max\ \{1 - w^T x + b, 0\}$. In addition, we would like $f(y) \leq -1$, $\forall y \in Y$. For a given y, this condition is equivalent to $1 + w^T y - b \leq 0$. Conversely, $1 + w^T y - b \geq 0$ indicates that $f(y) \geq -1$, such that the separation constraint is violated. Hence, this outcome should be penalised, e.g., via $l_Y(y) = \max\ \{1 + w^T y - b, 0\}$. Thus, for given $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$, the total penalty will be obtained as $\sum_{i=1}^N l_X(x_i) + \sum_{j=1}^M l_Y(y_j)$. Roughly speaking, minimising this function of w and b will yield a classifier $f(z) = w^T z - b$ with good accuracy on the training data at hand. However, we would also like this classifier to remain reasonably accurate even if small perturbations arise in the data, that is, we want it to be robust. From a geometric perspective, we know that $w^T z = b$ defines a hyperplane in $\mathbb{R}^n$. Recall that $w^T z = b + \epsilon$ is also a hyperplane, which can be obtained from the original one by translating it in the direction of w by an amount $\epsilon / ||w||_2^2 \in \mathbb{R}$. Indeed, $\forall z \in \mathbb{R}^n$ such that $w^T z = b$,

$$w^T(z + \frac{\epsilon}{||w||_2^2} w) = w^T z + \epsilon \frac{w^T w}{||w||_2^2} = b + \epsilon.$$

In other words, for any $b_1, b_2 \in \mathbb{R}$, the sets of points in $\mathbb{R}^n$ satisfying $w^T z = b_1$ and $w^T z = b_2$ correspond to two parallel hyperplanes separated by a distance $|b_2 - b_1| / ||w||_2^2$. In order to maximise the distance between these two hyperplanes, $||w||_2^2$ should therefore be as small as possible. In the case at hand, to increase classifier robustness, we would like to maximise the distance between hyperplanes $f(z) = 1$ and $f(z) = -1$, which can be achieved by selecting a vector $w \in \mathbb{R}^n$ whose 2-norm is small. The robustness criterion will therefore be expressed as the minimisation of $r(w) = ||w||_2^2$.

b. Recall that we are trying to balance classifier accuracy and robustness, that is, a trade-off between these two properties must be found. We therefore introduce a hyperparameter $\mu \in \mathbb{R}_+$ which will either favour robustness or accuracy. With this in mind, the classifier can be constructed by solving the following optimisation problem

$$\min_{w \in \mathbb{R}^n, b \in \mathbb{R}} r(w) + \mu \Big(\sum_{i=1}^N l_X(x_i) + \sum_{j=1}^M l_Y(y_j) \Big),$$

or, equivalently,

$$\min_{w \in \mathbb{R}^n, b \in \mathbb{R}} ||w||_2^2 + \mu \Big(\sum_{i=1}^N \max\ \{1 - w^T x_i + b, 0\} + \sum_{j=1}^M \max\ \{1 + w^T y_j - b, 0\} \Big).$$

c. The problem at hand can be readily recast as a second-order cone program. Indeed, let $t, s \in \mathbb{R}$, and let $u_i \in \mathbb{R}$, $i = 1, \ldots, N$, and $v_j \in \mathbb{R}$, $j = 1, \ldots, M$. Notice that minimising $||w||_2^2$ is equivalent to minimising t subject to $||w||_2^2 \leq st$ and $s = 1$. In addition, recall that minimising the max of two functions is equivalent to minimising a variable bounding these two functions simultaneously. In other words, the optimisation problem can be rewritten as

$$
\begin{aligned}
\min \ & t + \mu\Bigg(\sum_{i=1}^{N} u_i + \sum_{j=1}^{M} v_j \Bigg) \\
\text{s.t.} \ & ||w||_2^2 \leq st \\
& s = 1 \\
& 1 - w^T x_i + b \leq u_i, \ i = 1, \ldots, N \\
& 0 \leq u_i, \ i = 1, \ldots, N \\
& 1 + w^T y_j - b \leq v_j, \ j = 1, \ldots, M \\
& 0 \leq v_j, \ j = 1, \ldots, M \\
& w \in \mathbb{R}^n, \ b \in \mathbb{R} \\
& u_i \in \mathbb{R}, \ i = 1, \ldots, N \\
& v_j \in \mathbb{R}, \ j = 1, \ldots M.
\end{aligned}
$$

Note that the first inequality constraint defines a rotated quadratic cone, while all other constraints are linear inequalities. Since the objective function is linear, the problem at hand is indeed an instance of a second-order cone program.

Chapter 17

Semidefinite Programming

Solution to Problem 1

We seek a (tight) bound on the optimum via the dual problem. We start by rewriting the primal problem as

$$\begin{array}{ll} \min & 2p_1 + p_4 \\ \text{s.t.} & A_1p_1 + A_2p_2 + A_3p_3 + A_4p_4 - B \succeq 0, \end{array}$$

with

$$A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},\ A_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},\ A_3 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix},\ A_4 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The positive semidefiniteness constraint is now expressed as a *linear matrix inequality* (LMI). Let

$$b = \begin{pmatrix} 2 & 0 & 0 & 1 \end{pmatrix}^T \text{ and } p = \begin{pmatrix} p_1 & p_2 & p_3 & p_4 \end{pmatrix}^T.$$

In addition, let tr be a function associating the sum of its diagonal entries to any square matrix. Note that tr is a linear function. Indeed, for square matrices C and D of appropriate dimensions, $\text{tr}(C+D) = \text{tr}(C) + \text{tr}(D)$, and $\text{tr}(\alpha C) = \alpha \text{tr}(C)$, $\forall \alpha \in \mathbb{R}$. The trace also has other useful properties, such as

$$\text{tr}(C^TD) = \text{tr}(CD^T) = \text{tr}(D^TC) = \text{tr}(DC^T).$$

Using this function, we define the inner product $\text{tr}(U^TV) = \sum_{i,j} U_{ij}V_{ij}$ for matrices U and V of appropriate dimensions. We formulate the dual problem next. Since the cone of positive semidefinite matrices is self-dual, the dual variables can be viewed as the entries of a (symmetric) positive semidefinite matrix $X \in \mathbb{R}^{3\times 3}$. Now, let $L : \mathbb{R}^4 \times \mathbb{R}^{3\times 3} \mapsto \mathbb{R}$ be the Lagrangian function

$$L(p, X) = b^Tp - \text{tr}\Big(X^T\Big(\sum_{i=1}^4 A_ip_i - B\Big)\Big),$$

For any primal feasible p, the Lagrangian provides a lower bound on the primal objective, as

$$\text{tr}(ZY) = \text{tr}(ZY^{\frac{1}{2}}Y^{\frac{1}{2}}) = \text{tr}(Y^{\frac{1}{2}}ZY^{\frac{1}{2}}) \geq 0,$$

for any (symmetric) positive semidefinite matrices Z and Y of appropriate dimensions. The Lagrangian can also be successively re-expressed as

$$\begin{aligned} L(p, X) &= b^Tp - \text{tr}\Big(X^T\Big(\sum_{i=1}^4 A_ip_i - B\Big)\Big) \\ &= \sum_{i=1}^4 b_ip_i - \sum_{i=1}^4 \text{tr}\big(X^TA_ip_i\big) - \text{tr}\big(-X^TB\big) \end{aligned}$$

$$= \sum_{i=1}^{4} \Big(b_i - \mathrm{tr}\big(X^T A_i\big)\Big) p_i + \mathrm{tr}\big(X^T B\big).$$

Now, let $d : \mathbb{R}^{3\times 3} \mapsto \mathbb{R}$ be the dual function

$$d(X) = \inf_{p \in \mathbb{R}^4} L(p, X),$$

which obviously provides a lower bound on the primal objective. Since we seek finite lower bounds, we enforce $b_i - \mathrm{tr}(X^T A_i) = 0,\ i = 1, \ldots, 4$. The dual function therefore becomes $d(X) = \mathrm{tr}(X^T B)$, and the dual problem reads

$$\begin{array}{rl} \max & \mathrm{tr}(X^T B) \\ \text{s.t.} & \mathrm{tr}(X^T A_i) = b_i,\ i = 1, \ldots, 4 \\ & X \succeq 0. \end{array}$$

In the case at hand, substituting the expressions for $A_i,\ i = 1, \ldots, 4$, and $b_i,\ i = 1, \ldots, 4$, yields

$$\begin{array}{rl} \max & x_{22} \\ \text{s.t.} & x_{11} = 2 \\ & x_{12} + x_{21} = 0 \\ & x_{13} + x_{31} = 0 \\ & x_{22} + x_{33} = 1 \\ & X \succeq 0, \end{array}$$

with x_{ij} the entries of X. Since X is symmetric, $x_{ij} = x_{ji},\ i \neq j$. Hence, the second and third constraints imply that

$$x_{12} = x_{21} = x_{13} = x_{31} = 0.$$

In other words, the dual problem becomes

$$\begin{array}{rl} \max & x_{22} \\ \text{s.t.} & x_{22} + x_{33} = 1 \\ & \begin{pmatrix} 2 & 0 & 0 \\ 0 & x_{22} & x_{23} \\ 0 & x_{32} & x_{33} \end{pmatrix} \succeq 0. \end{array}$$

We then seek an algebraic criterion to express the positive semidefiniteness of X. Sylvester's criterion provides a necessary and sufficient condition to determine whether Hermitian or orthogonal matrices are positive definite. More precisely, positive definiteness can be asserted by checking that all leading principal minors are positive. To guarantee positive semidefiniteness, however, a stronger condition is required. Indeed, a sufficient condition for positive semidefiniteness is that all principal minors are nonnegative. Hence, applying this criterion to X yields

$$2 \geq 0,\ x_{22} \geq 0,\ x_{33} \geq 0,\ \begin{vmatrix} 2 & 0 \\ 0 & x_{22} \end{vmatrix} \geq 0,\ \begin{vmatrix} x_{22} & x_{23} \\ x_{32} & x_{33} \end{vmatrix} \geq 0,\ \begin{vmatrix} 2 & 0 \\ 0 & x_{33} \end{vmatrix} \geq 0,\ \begin{vmatrix} 2 & 0 & 0 \\ 0 & x_{22} & x_{23} \\ 0 & x_{32} & x_{33} \end{vmatrix} \geq 0.$$

Since X is symmetric, $x_{23} = x_{32}$, and the dual problem now reads

$$\begin{array}{rl} \max & x_{22} \\ \text{s.t.} & x_{22} + x_{33} = 1 \\ & x_{22}x_{33} - x_{23}^2 \geq 0 \\ & x_{22} \geq 0, x_{33} \geq 0, \end{array}$$

which is a conic program, as the second constraint defines a rotated quadratic cone. Since there exists a strictly feasible solution to this convex problem, e.g., $\begin{pmatrix} x_{22} & x_{33} & x_{23} \end{pmatrix} = \begin{pmatrix} 0.5 & 0.5 & 0 \end{pmatrix}$ is a feasible solution such that the nonlinear inequality is strictly satisfied, Slater's condition is verified and strong duality holds. Thus, at optimality, the primal and dual objectives are equal. In the case at hand, it is clear that the optimum is reached for $x_{22} = 1$, thus $2p_1 + p_4 = 1$. By virtue of the criterion for positive semidefiniteness, all diagonal entries of the matrix in the primal problem must be nonnegative. Hence, $p_4 - 1 \geq 0$, such that $p_1 = 0$ and $p_4 = 1$. It then follows that $p_2 = p_3 = 0$.

Solution to Problem 2

a. The problem of showing that a univariate polynomial p is nonnegative can be tackled by solving

$$p^\star = \min_{x\in\mathbb{R}} \quad p(x)$$

and checking that $p^\star \geq 0$. Alternatively, solving the equivalent maximisation problem

$$\begin{aligned} \rho^\star = \max_{\rho\in\mathbb{R}} \quad & \rho \\ \text{s.t.} \quad & p(x) - \rho \geq 0, \ \forall x \in \mathbb{R}, \end{aligned}$$

and checking that $\rho^\star \geq 0$ will also provide a *certificate of nonnegativity.* Note that in this equivalent problem, x is no longer a variable. Instead, the largest lower bound on p is sought. This problem therefore has a single variable, a linear objective as well as an (uncountably) infinite number of linear inequality constraints. Such problems are called *semi-infinite programs.* Although the problem at hand is linear, the infinite number of constraints makes it very challenging to solve. One way to overcome this issue consists in reformulating the constraints $p(x) - \rho \geq 0, \ \forall x \in \mathbb{R}$, so as to obtain a tractable convex problem. To this end, a popular approach consists in seeking to decompose the left-hand side expression $p(x) - \rho$ into a sum of squares. More precisely, let $2d$ denote the degree of p. Observe that p can only be of even degree, as odd degree polynomials cannot be nonnegative. In addition, let $z : \mathbb{R} \mapsto \mathbb{R}^{d+1}$, with $z(x)$ the vector of monomials $z(x) = \begin{pmatrix} 1 & x & \dots & x^d \end{pmatrix}^T$. Then, trying to express $p(x) - \rho$ as a sum of squares is equivalent to seeking $Q \in \mathbb{R}^{(d+1)\times(d+1)}$, $Q \succeq 0$, such that $p(x) - \rho = z(x)^T Q z(x), \ \forall x \in \mathbb{R}$. Clearly, if such a Q exists, $p(x) - \rho \geq 0, \ \forall x \in \mathbb{R}$. Thus, the equivalent problem becomes

$$\begin{aligned} \max_{\rho,Q} \quad & \rho \\ \text{s.t.} \quad & p(x) - \rho = z(x)^T Q z(x), \ \forall x \in \mathbb{R} \\ & Q \succeq 0, \end{aligned}$$

which is linear in the variables ρ and (the entries of) Q. Although the resulting problem might appear to be another instance of a semi-infinite program, it can be readily turned into a tractable convex program. Indeed, the constraints $p(x) - \rho = z(x)^T Q z(x), \ \forall x \in \mathbb{R}$, can be replaced by a finite set of linear equality constraints enforcing that the coefficients of the monomials on both sides of the original equality constraints are equal. In the case at hand, $z = \begin{pmatrix} 1 & x & x^2 \end{pmatrix}^T$, such that

$$\begin{aligned} z(x)^T Q z(x) &= \begin{pmatrix} 1 & x & x^2 \end{pmatrix} \begin{pmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{pmatrix} \begin{pmatrix} 1 \\ x \\ x^2 \end{pmatrix} \\ &= q_{11} + (q_{21} + q_{12})x + (q_{31} + q_{22} + q_{13})x^2 + (q_{23} + q_{32})x^3 + q_{33}x^4. \end{aligned}$$

Enforcing $p(x) - \rho = z(x)^T Q z(x), \ \forall x \in \mathbb{R}$, therefore yields

$$\begin{aligned} q_{11} &= 2 - \rho \\ q_{12} + q_{21} &= -2 \\ q_{31} + q_{22} + q_{13} &= 4 \\ q_{23} + q_{32} &= -2 \\ q_{33} &= 2, \end{aligned}$$

such that the following semidefinite program is obtained

$$\begin{aligned} \max \quad & 2 - q_{11} \\ \text{s.t.} \quad & q_{12} + q_{21} = -2 \\ & q_{31} + q_{22} + q_{13} = 4 \\ & q_{23} + q_{32} = -2 \\ & q_{33} = 2 \\ & Q \succeq 0. \end{aligned}$$

Now, let

$$B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \ A_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \ A_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \ A_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \ A_4 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

and $c = \begin{pmatrix} -2 & 4 & -2 & 2 \end{pmatrix}^T$. The semidefinite program can be expressed more concisely as

$$\begin{array}{ll} \max & 2 - \operatorname{tr}(B^T Q) \\ \text{s.t.} & \operatorname{tr}(A_i^T Q) = c_i, \ i = 1, \dots, 4 \\ & Q \succeq 0. \end{array}$$

Note that finding a solution to this problem implies that the polynomial p can be decomposed into a sum of squares, and is therefore nonnegative. In particular, it is worth mentioning that a nonnegative univariate polynomial of arbitrary (even) degree can always be decomposed into a sum of squares. Thus, in this case, the approximation is exact. The approximation is also exact for quadratic polynomials with an arbitrary number of variables, and for polynomials of degree four with two variables. However, in general, if the problem is infeasible, this only signifies that no such decomposition exists, and no conclusions can be drawn as to the nonnegativity of p.

b. The dual problem can be formed by introducing dual variables $p_i \in \mathbb{R}, \ i = 1, \dots, 4$, constructing the Lagrangian $L : \mathbb{R}^4 \times \mathbb{R}^{3\times 3} \mapsto \mathbb{R}$,

$$\begin{aligned} L(p, Q) &= 2 - \operatorname{tr}(B^T Q) + \sum_{i=1}^{4} p_i \Big(c_i - \operatorname{tr}(A_i^T Q) \Big) \\ &= -\operatorname{tr}\Big(\Big(B + \sum_{i=1}^{4} A_i p_i \Big)^T Q \Big) + 2 + \sum_{i=1}^{4} p_i c_i, \end{aligned}$$

and defining the dual function $d : \mathbb{R}^4 \mapsto \mathbb{R}$,

$$d(p) = \max_{Q \succeq 0} L(p, Q).$$

Since we are only interested in finite values of the dual function, the condition $B + \sum_{i=1}^4 A_i p_i \succeq 0$ must be enforced, such that $d(p) = 2 + \sum_{i=1}^4 p_i c_i$. The dual problem, which seeks to minimise the value of the dual function, therefore reads

$$\begin{array}{ll} \min & 2 + \sum_{i=1}^4 p_i c_i \\ \text{s.t.} & \sum_{i=1}^4 A_i p_i + B \succeq 0 \\ & p_i \in \mathbb{R}, \ i = 1, \dots, 4. \end{array}$$

Note that the first constraint is a linear matrix inequality. This dual problem can also be rewritten in a form similar to that of the primal problem in Problem 1,

$$\begin{array}{ll} \min & 2 - 2p_1 + 4p_2 - 2p_3 + 2p_4 \\ \text{s.t.} & \begin{pmatrix} 1 & p_1 & p_2 \\ p_1 & p_2 & p_3 \\ p_2 & p_3 & p_4 \end{pmatrix} \succeq 0. \end{array}$$

c. Although in theory solving a semidefinite program is preferable to tackling a semi-infinite program, only moderately-sized instances of general semidefinite programs can be efficiently solved in practice (as of 2020). Hence, seeking models with better computational properties is desirable, even if this often comes at the cost of imposing stricter conditions on the *ansatz* used to approximate p. In the sum-of-squares decomposition framework, the basic idea consists in finding algebraic conditions guaranteeing that $Q \succeq 0$ (without having to explicitly enforce it) and leading to more tractable optimisation problems, e.g., linear or second-order cone programs. For instance, instead of attempting to decompose p as a (general) sum of squares, a decomposition of p as a *scaled diagonally dominant sum of squares* could be sought. In such a context, the algebraic condition relies on the fact that a symmetric matrix $Q \in \mathbb{R}^{(d+1)\times(d+1)}$ is scaled diagonally dominant if and only if it can be expressed as

$$Q = \sum_{i<j} M^{ij},$$

for matrices $M^{ij} \in \mathbb{R}^{(d+1)\times(d+1)}$ having only four nonzero entries M^{ij}_{ii}, M^{ij}_{jj}, M^{ij}_{ij}, and M^{ij}_{ji} forming a symmetric and positive semidefinite 2×2 submatrix. Note that a scaled diagonally dominant matrix is positive semidefinite. Indeed, it is easy to check that $M^{ij} \succeq 0,\ i < j$, implying that

$$x^T Q x = x^T \left(\sum_{i<j} M^{ij} \right) x = \sum_{i<j} x^T M^{ij} x \geq 0, \ \forall x \in \mathbb{R}^d.$$

Note that the converse, however, is not true. In the case at hand, let

$$M^{12} = \begin{pmatrix} m^{12}_{11} & m^{12}_{12} & 0 \\ m^{12}_{21} & m^{12}_{22} & 0 \\ 0 & 0 & 0 \end{pmatrix}, \ M^{13} = \begin{pmatrix} m^{13}_{11} & 0 & m^{13}_{13} \\ 0 & 0 & 0 \\ m^{13}_{31} & 0 & m^{13}_{33} \end{pmatrix} \text{ and } M^{23} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & m^{23}_{22} & m^{23}_{23} \\ 0 & m^{23}_{32} & m^{23}_{33} \end{pmatrix}.$$

Hence, making use of the aforementioned algebraic condition leads to the following optimisation problem

$$\begin{array}{rl} \max & 2 - \mathrm{tr}(B^T Q) \\ \text{s.t.} & \mathrm{tr}(A_i^T Q) = c_i, \ i = 1, \ldots, 4 \\ & Q = M^{12} + M^{13} + M^{23} \\ & M^{12} \succeq 0, \ M^{13} \succeq 0, \ M^{23} \succeq 0. \end{array}$$

Recall that a sufficient condition for a matrix to be positive semidefinite is that its principal minors are nonnegative. Since matrices M^{ij} are symmetric and only have four nonzero entries, the semidefiniteness constraints simply reduce to

$$m^{ij}_{ii} m^{ij}_{jj} - (m^{ij}_{ij})^2 \geq 0, \ m^{ij}_{ii} \geq 0, \text{ and } m^{ij}_{jj} \geq 0,$$

and define rotated quadratic cones in m^{ij}_{ii}, m^{ij}_{jj} and m^{ij}_{ij}. Hence, the conic program is obtained

$$\begin{array}{rl} \max & 2 - q_{11} \\ \text{s.t.} & q_{12} + q_{21} = -2 \\ & q_{31} + q_{22} + q_{13} = 4 \\ & q_{23} + q_{32} = -2 \\ & q_{33} = 2 \\ & q_{11} = m^{12}_{11} + m^{13}_{11} \\ & q_{12} = m^{12}_{12} \\ & q_{12} = q_{21} \\ & q_{13} = m^{13}_{13} \\ & q_{13} = q_{31} \\ & q_{22} = m^{12}_{22} + m^{23}_{22} \\ & q_{23} = m^{23}_{23} \\ & q_{23} = q_{32} \\ & q_{33} = m^{13}_{33} + m^{23}_{33} \\ & m^{12}_{11} m^{12}_{22} - (m^{12}_{12})^2 \geq 0 \\ & m^{13}_{11} m^{13}_{33} - (m^{13}_{13})^2 \geq 0 \\ & m^{23}_{22} m^{23}_{33} - (m^{23}_{23})^2 \geq 0 \\ & m^{12}_{11} \geq 0, \ m^{13}_{11} \geq 0 \\ & m^{12}_{22} \geq 0, \ m^{23}_{22} \geq 0 \\ & m^{13}_{33} \geq 0, \ m^{23}_{33} \geq 0, \end{array}$$

which can be readily turned into a second-order cone program.

d. Recall that a real symmetric matrix $Q \in \mathbb{R}^{(d+1)\times(d+1)}$ with nonnegative diagonal entries is diagonally dominant if

$$q_{ii} \geq \sum_{j\neq i} |q_{ij}|,\ i = 1, \ldots, d+1.$$

By virtue of Gershgorin's circle theorem, real symmetric diagonally dominant matrices with nonnegative diagonal entries are also positive semidefinite. Indeed, let $\Lambda(Q) = \{\lambda \in \mathbb{R} | \det(\lambda I - Q) = 0\}$ be the set of eigenvalues of Q. In addition, let $D_i(Q) = \{z \in \mathbb{C} | |z - q_{ii}| \leq \sum_{j\neq i} |q_{ij}|\},\ i = 1, \ldots, d+1$, be a set of disks in the complex plane whose centres are the diagonal entries q_{ii} of Q and whose radii are $R_i = \sum_{j\neq i} |q_{ij}|,\ i = 1, \ldots, d+1$. Then, Gershgorin's circle theorem essentially states that

$$\Lambda(Q) \subset \bigcup_{i\in\{1,\ldots,d+1\}} D_i(Q).$$

Moreover, for Q diagonally dominant,

$$|\lambda - q_{ii}| \leq \sum_{j\neq i} |q_{ij}| \leq q_{ii},\ \forall\lambda \in D_i(Q),\ i = 1, \ldots, d+1.$$

Since symmetric matrices have real eigenvalues, the inequalities above imply that $\lambda \geq 0$. In other words, the eigenvalues of Q are nonnegative and the matrix is therefore positive semidefinite. Hence, the algebraic conditions above can be used to check whether a polynomial p can be decomposed into a *diagonally dominant sum of squares*. For each $i = 1, \ldots, d+1$, introducing variables $z_{ij} \in \mathbb{R},\ j = 1, \ldots, d+1,\ j \neq i$, allows for the reformulation of the algebraic conditions in terms of linear inequality constraints,

$$\begin{aligned} q_{ii} &\geq \sum_{j\neq i} z_{ij} \\ -z_{ij} &\leq q_{ij} \leq z_{ij},\ j = 1, \ldots, d+1,\ j \neq i. \end{aligned}$$

Thus, a linear program can be formulated to check whether p can be expressed as a diagonally dominant sum of squares. In the case at hand, this linear program reads

$$\begin{array}{rl} \max & 2 - q_{11} \\ \text{s.t.} & q_{12} + q_{21} = -2 \\ & q_{31} + q_{22} + q_{13} = 4 \\ & q_{23} + q_{32} = -2 \\ & q_{33} = 2 \\ & q_{12} = q_{21} \\ & q_{13} = q_{31} \\ & q_{23} = q_{31} \\ & q_{11} \geq z_{12} + z_{13} \\ & -z_{12} \leq q_{12} \leq z_{12} \\ & -z_{13} \leq q_{13} \leq z_{13} \\ & q_{22} \geq z_{21} + z_{23} \\ & -z_{21} \leq q_{21} \leq z_{21} \\ & -z_{23} \leq q_{23} \leq z_{23} \\ & q_{33} \geq z_{31} + z_{32} \\ & -z_{31} \leq q_{31} \leq z_{31} \\ & -z_{32} \leq q_{32} \leq z_{32} \\ & q_{11} \geq 0,\ q_{22} \geq 0,\ q_{33} \geq 0. \end{array}$$

Note that despite the computational advantages of linear and second-order cone programming formulations, stricter conditions have been imposed on the structure of the decomposition of p to derive them. Let DSOS_{2d}, SDSOS_{2d} and SOS_{2d} denote the sets of (univariate) polynomials of degree $2d$ that may be decomposed into a diagonally dominant sum of squares, a scaled diagonally dominant sum of squares and a (general) sum of squares, respectively. In addition, let PSD_{2d} be the set of nonnegative (univariate) polynomials of degree $2d$. Then, in the particular case of univariate polynomials, it can be shown that $\text{DSOS}_{2d} \subseteq \text{SDSOS}_{2d} \subseteq \text{SOS}_{2d} = \text{PSD}_{2d}$. Note that in general, however, the last relation will be replaced by $\subseteq$. Hence, nothing can be said as to the nonnegativity of p if the linear and second-order cone programs happen to be infeasible.

Chapter 18

Unconstrained Optimization and Descent Methods

Solution to Problem 1

a. Recall that *descent methods* seek an approximate minimiser of some continuously differentiable function $f : \mathbb{R}^n \mapsto \mathbb{R}$ by producing a sequence of iterates $\{z^k\}_{k\in\mathbb{N}}$ of the form $z^{k+1} = z^k + t^k d^k$, with $t^k > 0$ the step size and d^k a so-called *descent direction*, i.e., such that $(d^k)^T \nabla f(z_k) < 0$. Note that the gradient $\nabla f(z^k)$ evaluated at z^k points in the direction of steepest local ascent, and any vector $d^k \in \mathbb{R}^n$ satisfying the aforementioned condition will therefore point in a direction in which f decreases. Although a variety of descent directions can be envisaged, the so-called gradient descent direction consists in taking $d^k = -\nabla f(z^k)$. Clearly, $(d^k)^T \nabla f(z^k) = -\nabla f(z^k)^T \nabla f(z^k) = -||\nabla f(z^k)||_2^2 < 0$, unless z^k is a stationary point of f. This direction is sometimes referred to as the *steepest descent direction.* In the case at hand, the gradient of g can be readily computed as

$$\nabla f = \begin{pmatrix} \cos(x+y) + \frac{x}{3} \\ \cos(x+y) + y \end{pmatrix},$$

and evaluating it at (x_0, y_0) yields

$$\nabla f(x_0, y_0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

The gradient descent direction is immediately obtained as $d_G = -\nabla f(x_0, y_0) = \begin{pmatrix} -1 & -1 \end{pmatrix}^T$. Note that the gradient descent method is referred to as a *first-order method* as it only exploits information about the first derivatives of f or g.

b. Recall that in the context of unconstrained optimisation problems, Newton's method seeks a stationary point of a twice continuously differentiable function $f : \mathbb{R}^n \mapsto \mathbb{R}$ by solving the system of (usually nonlinear) equations $\nabla f(z) = 0$. The Jacobian of this system is the Hessian matrix $H \in \mathbb{R}^{n\times n}$, which has entries $H_{ij} = \partial f / \partial z_i z_j$. Note that $H = H^T$, as f is assumed twice continuously differentiable. Provided that $H(z^k)$ is not singular, in its simplest form, Newton's method produces a sequence of iterates

$$z^{k+1} = z^k - H^{-1}(z^k)\nabla f(z^k),$$

with $H^{-1}(z^k)$ the inverse of the Hessian evaluated at z^k. This update rule can be obtained from the generic rule given in **a.** by taking steps of constant length $t^k = 1$ and following direction $d^k = -H^{-1}(z^k)\nabla f(z^k)$. Thus, Newton's method can be viewed as scaled version of the classical gradient descent method. Observe that d_k is only a valid descent direction if $H(z^k)$ is positive definite. Indeed, $x^T H(z^k) x > 0$, $\forall x \in \mathbb{R}^n$, implies that $0 < x^T H(z^k) x = x^T H(z^k) H^{-1}(z^k) H(z^k) x = y^T H^{-1}(z^k) y$, $\forall y \in \mathbb{R}^n$, thus $(d^k)^T \nabla f(z^k) = -\nabla f(z^k) H^{-1}(z^k) \nabla f(z^k) < 0$. In the case at hand, the Hessian can be readily formed as

$$H(x,y) = \begin{pmatrix} -\sin(x+y) + \frac{1}{3} & -\sin(x+y) \\ -\sin(x+y) & -\sin(x+y) + 1 \end{pmatrix},$$

such that

$$H(x_0, y_0) = \begin{pmatrix} 1/3 & 0 \\ 0 & 1 \end{pmatrix} \succ 0 \text{ and } H^{-1}(x_0, y_0) = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \succ 0.$$

Then, the Newton direction d_N can be computed as

$$d_N = -H^{-1}(x_0, y_0)\nabla f(x_0, y_0) = -\begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -3 \\ -1 \end{pmatrix}.$$

Note that setting $t^0 = 0.3131$ and taking $z^1 = z^0 + t^0 d_N = \begin{pmatrix} -0.9393 & -0.3131 \end{pmatrix}^T$ yields a local minimiser of g, i.e., Newton's method converges in one step. The gradient method would have required more than one step. Finally, it is worth noting that in contrast to the gradient descent method, Newton's method makes use of second-order derivatives of f, and is therefore called a *second-order method.*

Solution to Problem 2

Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be a continuously differentiable function. Recall that the gradient descent method consists in constructing a sequence of iterates $\{z^k\}_{k\in\mathbb{N}}$ such that $z^{k+1} = z^k + t^k d^k$, for some appropriate step size $t^k > 0$ and descent direction $d^k = -\nabla f(z^k)$. Thus, forming d^k requires the computation and evaluation of all first-order derivatives of f at each iteration, which may sometimes prove expensive or impractical. In fact, most of the time, computing the full gradient at each iteration is not required to obtain a valid descent direction. Indeed, let $i_k \in \{1, \ldots, n\}$ be an appropriately selected coordinate index, let $[\nabla f]_{i_k} = \partial f / \partial z_{i_k}$ denote the i_k^{th} entry of the gradient of f evaluated at z^k, and let $e_{i_k} \in \mathbb{R}^n$ be a vector whose entries are all equal to 0, except the i_k^{th}, which is equal to 1. Then, let $d^k = -[\nabla f(z^k)]_{i_k} e_{i_k}$, and observe that $(d^k)^T \nabla f(z^k) = -[\nabla f(z^k)]_{i_k} e_{i_k}^T \nabla f(z^k) = -[\nabla f(z^k)]_{i_k}^2 < 0$, unless z^k is a coordinate-wise minimiser of f, i.e., $f(z_k) \leq f(z_k + te_{i_k})$, $\forall t \in \mathbb{R}$. In other words, d^k is a valid descent direction most of the time, and it only requires the computation of one partial derivative of f along coordinate i_k. Algorithms resorting to such descent directions are usually referred to as *coordinate descent methods.* In their simplest form, a coordinate index i_k is selected at each iteration $k \in \mathbb{N}$, e.g. by cycling among coordinates such that $i_{k+1} = [i_k \mod n] + 1$, the associated coordinate descent direction is computed, and the next iterate is formed with the classical update rule. Such methods therefore select directions that *alternate* between coordinates in an attempt to identify a minimiser of f. Note that in such a setup, the step size t^k can often be computed by exact minimisation of f along d^k (an exact line search), which simply reduces to the one dimensional problem

$$t^k = \arg\min_{t\geq 0} f(z^k + td^k) = \arg\min_{t\geq 0} f(z^k - t[\nabla f(z^k)]_{i_k} e_{i_k}).$$

There is much scope for variation with such methods. In particular, *block coordinate descent methods* constitute a straightforward extension of the aforementioned techniques, whereby a subset of indices is selected at each iteration and the coordinates of the next iterate are updated sequentially. More precisely, let $I_k = \{i_1, \ldots, i_M\} \subset \{1, \ldots, n\}$ denote the set of (non-repeated) coordinate indices selected at iteration k, such that $1 \leq i_1 < i_2 < \ldots < i_M \leq n$. Then, the coordinates of the next iterate z^{k+1} are computed as

$$\begin{aligned}
z_{i_1}^{k+1} &= z_{i_1}^k - t_{i_1}^k [\nabla f(z_1^k, \ldots, z_n^k)]_{i_1} \\
z_{i_2}^{k+1} &= z_{i_2}^k - t_{i_2}^k [\nabla f(z_1^k, \ldots, z_{i_1}^{k+1}, \ldots, z_n^k)]_{i_2} \\
&\vdots \\
z_{i_M}^{k+1} &= z_{i_M}^k - t_{i_M}^k [\nabla f(z_1^k, \ldots, z_{i_1}^{k+1}, \ldots, z_{i_{M-1}}^{k+1}, \ldots, z_n^k)]_{i_M}
\end{aligned}$$

for appropriately selected step sizes $t_i^k > 0$, $\forall i \in I_k$, and

$$z_i^{k+1} = z_i^k, \ \forall i \in \{1, \ldots, n\} \setminus I_k,$$

respectively. Block coordinate methods are particularly attractive when the objective function is amenable to a sum of functions of subsets of coordinates of z^k, i.e., when the objective has a *separable* structure. Indeed, in such a context, the aforementioned subsets of coordinates define the so-called blocks, and block coordinate updates can be performed independently and thus parallelised, which may bring about significant computational benefits. It is also worth mentioning that these methods are directly connected to *stochastic gradient descent methods* typically used in machine learning applications.

In the case at hand, $g(z_1, z_2) = \frac{(z_1+z_2-2)^2}{2} + (z_1 - z_2 + 2)^2$, and let $z^0 = (z_1^0, z_2^0) = (0, 0)$. The partial derivatives of g with respect to z_1 and z_2 write

$$\frac{\partial g}{\partial z_1} = (z_1+z_2-2)+2(z_1-z_2+2) = 3z_1 - z_2 + 2 \text{ and } \frac{\partial g}{\partial z_2} = (z_1+z_2-2)-2(z_1-z_2+2) = -z_1+3z_2-6.$$

In the block coordinate framework using an exact line search, the coordinates of the next iterate are obtained as

$$z_1^1 = \arg\min_x \; g(x, z_2^0)$$
$$z_2^1 = \arg\min_y \; g(z_1^1, y).$$

Writing the optimality conditions of the first subproblem allows us to retrieve the value of z_1^1,

$$\left.\frac{\partial g}{\partial z_1}\right|_{(z_1^1, z_2^0)} = 0 \Leftrightarrow 3z_1^1 - z_2^0 + 2 = 0 \Rightarrow z_1^1 = -\frac{2}{3}.$$

Likewise, expressing the optimality conditions of the second subproblem yields

$$\left.\frac{\partial g}{\partial z_2}\right|_{(z_1^1, z_2^1)} = 0 \Leftrightarrow -z_1^1 + 3z_2^1 - 6 = 0 \Rightarrow z_2^1 = \frac{16}{9},$$

such that $z^1 = (z_1^1, z_2^1)^T = (-2/3, 16/9)^T$. The coordinates of the second iterate can be readily computed as

$$3z_1^2 - z_2^1 + 2 = 0 \Rightarrow z_1^2 = \frac{-2}{27} \text{ and } -z_1^2 + 3z_2^2 - 6 = 0 \Rightarrow z_2^2 = 2 + \frac{2}{81},$$

and $z^2 = (z_1^2, z_2^2)^T = (-2/27, 2 + 2/81)^T$. Coding up this procedure shows that a handful of iterations are required to converge to $z^\star = (z_1^\star, z_2^\star)^T = (0, 2)^T$. It is straightforward to check that $z^\star$ is a stationary point corresponding to a (global) minimum of g, as $g(z^\star) = 0 \le g(z), \forall z \in \mathbb{R}^2$. Finally, even though the coordinate descent method invoked in this problem quickly converged to a stationary point, it is worth noting that it may not always be the case, even for convex functions. Relatively strong conditions are typically required, e.g., on the smoothness of the objective (and its derivatives), to guarantee that coordinate descent methods indeed converge to stationary points.

Solution to Problem 3

Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be a continuously differentiable function. Recall that the gradient descent algorithm is an iterative method seeking approximate minimisers of f by taking a step in the steepest (local) descent direction at each iteration. More formally, the method produces a sequence of iterates $\{z^k\}_{k\in\mathbb{N}}$ via the update rule $z^{k+1} = z^k + t^k d^k$, with step size $h_k > 0$ and direction $d^k = -\nabla f(z^k)$. There are several ways of selecting the step size t^k at each iteration. For instance, an *exact line search* could be performed along d^k, that is,

$$t^k = \arg\min_{t\ge 0} \; f(z^k + td^k).$$

Although this method of selecting step sizes guarantees that the gradient descent method converges to a stationary point of f, solving this subproblem at each iteration is computationally expensive. Hence, this approach usually proves impractical. Instead, *inexact line search* methods have been proposed. Roughly speaking, inexact line search methods attempt to cheaply identify step sizes that achieve a sufficiently large decrease in objective function value at each iteration, while ensuring convergence of the algorithm. These specifications are usually encoded by algebraic conditions that can be easily checked, and the candidate step size is updated iteratively until a step size satisfying all conditions is found. An instance of such algebraic conditions is the Wolfe conditions,

$$f(z^k + t^k d^k) \le f(z^k) + \beta_1 t^k (d^k)^T \nabla f(z^k),$$

and

$$(d^k)^T \nabla f(z^k + t^k d^k) \geq \beta_2 (d^k)^T \nabla f(z^k),$$

with $0 < \beta_1 < \beta_2 < 1$. The parameters β_1 and β_2 are often selected such that $0 < \beta_1 < 0.5$ and $0.5 < \beta_2 < 1$. Roughly speaking, the first condition, which is also known as *Armijo's condition*, ensures that the decrease in objective function value is sufficiently big, and implicitly defines an upper bound on t^k. The second condition, which is sometimes referred to as the *curvature condition*, guarantees that the step size is not too small.

Now, let $t_L = 0$ and $t_H = +\infty$. The inexact line search method developed in the following essentially seeks a value t^k such that $t_L < t^k < t_H$ and the Wolfe conditions are satisfied. This is achieved by progressively updating t^k as well as the bounds t_L and t_H. Let $t^k = 1$, $\beta_1 = 1/4$ and $\beta_2 = 3/4$ and let us assume that the current iterate is $z^k = (x^k, y^k) = (0, 0)$. In order to evaluate the Wolfe conditions and form d^k, the expression of the gradient of g must be computed,

$$\nabla g(x, y) = \begin{pmatrix} 4x^3 - 4xy + 2x - 2 \\ 2y - 2x^2 \end{pmatrix},$$

such that $d^k = -\nabla f(z^k) = -\begin{pmatrix} -2 & 0 \end{pmatrix}^T = \begin{pmatrix} 2 & 0 \end{pmatrix}^T$. The candidate iterate can then be computed as

$$\hat{z}^{k+1} = z^k + t^k d^k = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + 1 \begin{pmatrix} 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix},$$

such that $f(z^k) = 1$, $f(\hat{z}^{k+1}) = 17$ and $\beta_1 t^k (d^k)^T \nabla g(z^k) = (1/4) \times 1 \times (-4) = -1$. Armijo's condition is violated, as $17 > 1 + (-1) = 0$, which implies that the step size is too big. The upper bound is therefore updated such that $t_H = 1$. In this case, a bisection type update rule is used to compute the new candidate step size, which becomes $t^k = (t_L + t_H)/2 = 1/2$. The candidate iterate must also be updated accordingly

$$\hat{z}^{k+1} = z^k + t^k d^k = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

which yields $f(\hat{z}^{k+1}) = 1$ and $\beta_1 t^k (d^k)^T \nabla g(z^k) = (1/4) \times (1/2) \times (-4) = -1/2$. The first Wolfe condition is again violated, as $1 > 1 + (-1/2) = 1/2$. The upper bound thus becomes $t_H = 1/2$, while $t^k = (t_L + t_H)/2 = 1/4$. The candidate iterate must be update once more,

$$\hat{z}^{k+1} = z^k + t^k d^k = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \frac{1}{4} \begin{pmatrix} 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 0 \end{pmatrix},$$

such that $f(\hat{z}^{k+1}) = (1/16) + (1/4) = 5/16$ and $\beta_1 t^k (d^k)^T \nabla g(z^k) = (1/4) \times (1/4) \times (-4) = -1/4$. This time, Armijo's condition is satisfied, as $5/16 \leq 1 + (-1/4) = 3/4 = 12/16$. The second condition must now be checked. Evaluating the numerator and denominator of the expression appearing on the left-hand side yields $(d^k)^T \nabla g(\hat{z}^{k+1}) = -1$ and $(d^k)^T \nabla g(z^k) = -4$, such that $-1 \geq (3/4) \times (-4) = -3$ and the second Wolfe condition is also satisfied. In other words, $t^k = 1/4$ is a valid step size, and taking a gradient step therefore yields the next iterate $z^{k+1} = \hat{z}^{k+1} = \begin{pmatrix} 1/2 & 0 \end{pmatrix}^T$.

Chapter 19

Constrained Optimization and Interior Point Methods

Solution to Problem 1

To be added.

Chapter 20

Automatic Differentiation

Solution to Problem 1

a. Computing gradients or higher-order derivatives of the objective function is required in a broad class of optimisation algorithms. To this end, several techniques can be invoked, including symbolic differentiation or finite differences. In particular, if we have an analytical expression of the objective function at hand, and provided that it is relatively simple, the expression of the gradient or that of higher-order derivatives can often be computed by hand or via a software performing symbolic calculations. Once the expression of the gradient is available, computing its value then requires n function evaluations, with n the number of optimisation variables. In some cases, however, although an analytical formula may be available, it may be too cumbersome to resort to symbolic calculations. On the other hand, if no analytical expression is available or the objective function is only available in the form of computer code, it may be necessary to employ finite difference schemes, which approximate derivatives numerically. In this setup, computing a gradient with a classical (first-order) finite difference scheme requires $n+1$ function evaluations. This technique may unfortunately be inaccurate, mostly as a result of the accumulation of roundoff errors over time. Worse, it may also prove numerically unstable. None of the techniques described above are completely satisfactory. An algorithmic solution, named *automatic differentiation*, has therefore been proposed to compute derivatives exactly and efficiently in a variety of settings.

Roughly speaking, in the context of automatic differentiation, the function to differentiate is treated as a composition of simpler functions, down to the level of elementary arithmetic operations and functions, e.g., exponentials, logarithms or sines. The chain rule is then employed to progressively compute inner derivatives and partial derivatives of the function of interest with respect to optimisation variables are eventually obtained. Two automatic differentiation strategies exist, namely *forward* and *reverse accumulation*, which are also known as the *forward mode* and the *reverse mode*, respectively. Forward accumulation essentially starts "from the inside of the chain rule" and makes its way "towards the outside", whereas reverse accumulation does the exact opposite. For example, let F be a function of x obtained as the composition of functions g, h, and p, that is, $F(x) = F(g(h(p(x))))$. By the chain rule, the partial derivative of F with respect to x is

$$\frac{\partial F}{\partial x} = \frac{\partial F}{\partial g}\frac{\partial g}{\partial h}\frac{\partial h}{\partial p}\frac{\partial p}{\partial x}.$$

The forward accumulation technique starts by computing

$$\frac{\partial p}{\partial x}, \text{ and proceeds with } \frac{\partial h}{\partial x} = \frac{\partial h}{\partial p}\frac{\partial p}{\partial x}, \text{ and } \frac{\partial g}{\partial x} = \frac{\partial g}{\partial h}\frac{\partial h}{\partial x},$$

until the partial derivative of F with respect to x is reached. By contrast, the reverse accumulation technique begins with

$$\frac{\partial F}{\partial g}, \text{ then moves on to } \frac{\partial F}{\partial h} = \frac{\partial F}{\partial g}\frac{\partial g}{\partial h} \text{ and } \frac{\partial F}{\partial p} = \frac{\partial F}{\partial h}\frac{\partial h}{\partial p},$$

until the partial derivative of F with respect to x has been computed. Note that the forward accumulation technique is particularly well-suited for functions with relatively few arguments and many outputs, while the opposite holds true for reverse accumulation. Indeed, only the innermost partial derivatives will change for functions with few outputs and many arguments. Finally, it is also worth mentioning that a direct connection exists between the *backpropagation* algorithm at the heart of deep learning/neural networks and the reverse accumulation approach.

In the case at hand, the function f must be decomposed into a set of elementary functions. To this end, a set of symbols representing the arguments of these functions is introduced. In particular, let $w_0 = x$, $w_1 = y$, $w_2 = w_0^2$, $w_3 = w_0 w_1$, $w_4 = \sin(w_3)$, $w_5 = w_4/w_0$ and $w_6 = w_2 + w_5 = f$. A crucial device on which automatic differentiation methods rely is the so-called *computation* or *computational graph*. The computational graph, which is directed and acyclic, essentially encodes the relationships between the different functions (and their arguments) composing f. More specifically, each edge represents a function argument, and a node is a function of its incoming edges. Inspecting the structure of the computational graph greatly simplifies the computation of inner partial derivatives, as discussed below. The computational graph of the problem at hand is shown in Figure 20.1.

Now, in order to obtain the gradient of f, the partial derivatives of f with respect to both x and y must be computed. We start with the partial derivative of f with respect to x. Let $w_0' = \partial w_0/\partial x = 1$ and $w_1' = \partial w_1/\partial x = 0$. In what follows, the prime symbol $'$ will be used to denote a partial derivative with respect to x. Then, following the edges of the computational graph in topological order (from top to bottom in Figure 20.1) and applying the forward accumulation technique successively yields

$$\begin{aligned}
w_2' &= \frac{\partial w_2}{\partial w_0}\frac{\partial w_0}{\partial x} = 2w_0 w_0' = 2w_0 \\
w_3' &= \frac{\partial w_3}{\partial w_0}\frac{\partial w_0}{\partial x} + \frac{\partial w_3}{\partial w_1}\frac{\partial w_1}{\partial x} = w_1 w_0' + w_0 w_1' = w_1 \\
w_4' &= \frac{\partial w_4}{\partial w_3}\frac{\partial w_3}{\partial x} = \frac{\partial w_4}{\partial w_3} w_3' = \cos(w_3) w_1 \\
w_5' &= \frac{\partial w_5}{\partial w_4}\frac{\partial w_4}{\partial x} + \frac{\partial w_5}{\partial w_0}\frac{\partial w_0}{\partial x} = \frac{\partial w_5}{\partial w_4} w_4' + \frac{\partial w_5}{\partial w_0} w_0' = \frac{1}{w_0}\cos(w_3) w_1 - \frac{w_4}{w_0^2} \\
w_6' &= \frac{\partial w_6}{\partial w_2}\frac{\partial w_2}{\partial x} + \frac{\partial w_6}{\partial w_5}\frac{\partial w_5}{\partial x} = \frac{\partial w_6}{\partial w_2} w_2' + \frac{\partial w_6}{\partial w_5} w_5' = 2w_0 + \frac{w_1}{w_0}\cos(w_3) - \frac{w_4}{w_0^2} = \frac{\partial f}{\partial x}.
\end{aligned}$$

The partial derivative of f with respect to y must now be computed. Let $\dot{w}_0 = \partial w_0/\partial y = 0$ and $\dot{w}_1 = \partial w_1/\partial y = 1$. In the following, the dot symbol $\dot{}$ will be used to indicate a partial derivative with respect to y. Then, resorting to the forward accumulation strategy leads to

$$\begin{aligned}
\dot{w}_2 &= \frac{\partial w_2}{\partial w_0}\frac{\partial w_0}{\partial y} = \frac{\partial w_2}{\partial w_0}\dot{w}_0 = 0 \\
\dot{w}_3 &= \frac{\partial w_3}{\partial w_0}\frac{\partial w_0}{\partial y} + \frac{\partial w_3}{\partial w_1}\frac{\partial w_1}{\partial y} = \frac{\partial w_3}{\partial w_0}\dot{w}_0 + \frac{\partial w_3}{\partial w_1}\dot{w}_1 = 0 + w_0 = w_0 \\
\dot{w}_4 &= \frac{\partial w_4}{\partial w_3}\frac{\partial w_3}{\partial y} = \frac{\partial w_4}{\partial w_3}\dot{w}_3 = \cos(w_3) w_0 \\
\dot{w}_5 &= \frac{\partial w_5}{\partial w_4}\frac{\partial w_4}{\partial y} + \frac{\partial w_5}{\partial w_0}\frac{\partial w_0}{\partial y} = \frac{\partial w_5}{\partial w_4}\dot{w}_4 + \frac{\partial w_5}{\partial w_0}\dot{w}_0 = \frac{1}{w_0}\cos(w_3) w_0 + 0 = \cos(w_3) \\
\dot{w}_6 &= \frac{\partial w_6}{\partial w_5}\frac{\partial w_5}{\partial y} + \frac{\partial w_6}{\partial w_2}\frac{\partial w_2}{\partial y} = \frac{\partial w_6}{\partial w_5}\dot{w}_5 + \frac{\partial w_6}{\partial w_2}\dot{w}_2 = \cos(w_3) + 0 = \cos(w_3) = \frac{\partial f}{\partial y}.
\end{aligned}$$

The gradient can now be formed from the partial derivatives of f with respect to x and y. Note that the number of terms appearing in the computation of inner derivatives is equal to the number of edges incident to the node whose output represents the relevant function in the computational graph.

b. Let us now apply the reverse accumulation technique. Let $f = w_6$ and $\bar{w}_6 = \partial f/\partial w_6 = 1$. In the upcoming developments, the bar symbol $\bar{}$ will be used to denote a partial derivative of f with respect to the symbol underneath the bar, i.e., $\bar{w}_i = \partial f/\partial w_i$. Following the edges of the computational graph in reverse topological order (from bottom to top in Figure 20.1), we successively find

$$\begin{aligned}
\bar{w}_5 &= \frac{\partial f}{\partial w_6}\frac{\partial w_6}{\partial w_5} = \bar{w}_6\frac{\partial w_6}{\partial w_5} = 1 \\
\bar{w}_4 &= \frac{\partial f}{\partial w_5}\frac{\partial w_5}{\partial w_4} = \bar{w}_5\frac{\partial w_5}{\partial w_4} = \frac{1}{w_0}
\end{aligned}$$

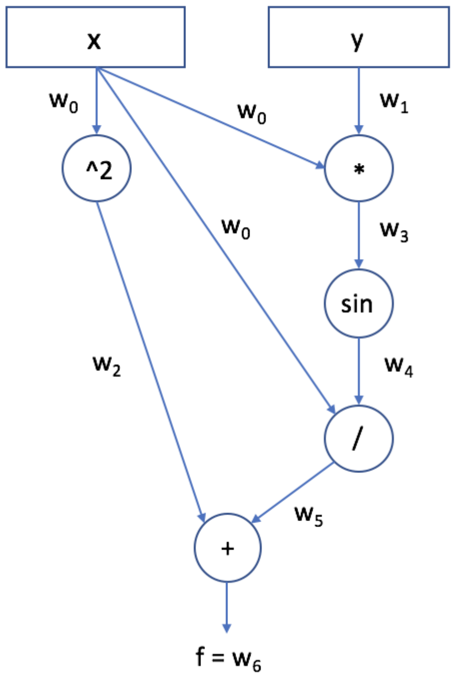


Figure 20.1: Computational graph of the problem at hand. In nodes, the caret represents the exponent operator (here a square), the star denotes a product, the sin represents the sine function, the slash stands for a division, and the plus describes an addition.

$$
\begin{aligned}
\bar{w}_3 &= \frac{\partial f}{\partial w_4}\frac{\partial w_4}{\partial w_3} = \bar{w}_4\frac{\partial w_4}{\partial w_3} = \frac{1}{w_0}\cos(w_3)\\
\bar{w}_2 &= \frac{\partial f}{\partial w_6}\frac{\partial w_6}{\partial w_2} = \bar{w}_6\frac{\partial w_6}{\partial w_2} = 1\\
\bar{w}_1 &= \frac{\partial f}{\partial w_3}\frac{\partial w_3}{\partial w_1} = \bar{w}_3\frac{\partial w_3}{\partial w_1} = \frac{1}{w_0}\cos(w_3)w_0 = \cos(w_3) = \frac{\partial f}{\partial y}\\
\bar{w}_0 &= \frac{\partial f}{\partial w_2}\frac{\partial w_2}{\partial w_0} + \frac{\partial f}{\partial w_3}\frac{\partial w_3}{\partial w_0} + \frac{\partial f}{\partial w_5}\frac{\partial w_5}{\partial w_0} = \bar{w}_2\frac{\partial w_2}{\partial w_0} + \bar{w}_3\frac{\partial w_3}{\partial w_0} + \bar{w}_5\frac{\partial w_5}{\partial w_0} = 2w_0 + \frac{w_1}{w_0}\cos(w_3) - \frac{w_4}{w_0^2} = \frac{\partial f}{\partial x},
\end{aligned}
$$

and the gradient can be readily computed. Unsurprisingly, the same results as those obtained via forward accumulation are found. It is also straightforward to verify that symbolic calculations would indeed yield these results as well.