0.1 Linear Optimization

In this section, we study on linear optimization problems, i.e., linear programs (LPs).

0.1.1. Problem Formulation

Remember, for a linear program (LP), we want to maximize or minimize a linear **objective function** of the continous decision variables, while considering linear constraints on the values of the decision variables.

Definition 0.1: Linear Function

function $f(x_1, x_2, \cdots, x_n)$ is linear if, and only if, we have $f(x_1, x_2, \cdots, x_n) = c_1x_1 + c_2x_2 + \cdots + c_nx_n$, where the $c_1, c_2, \cdots, c_n$ coefficients are constants.

A Generic Linear Program (LP)

Decision Variables:
x_i : continuous variables ($x_i \in \mathscr{R}$, i.e., a real number), $\forall i = 1, \cdots, 3$.

Parameters (known input parameters):
c_i : cost coefficients $\forall i = 1, \cdots, 3$
a_{ij} : constraint coefficients $\forall i = 1, \cdots, 3,\ j = 1, \cdots, 4$
b_j : right hand side coefficient for constraint j, $j = 1, \cdots, 4$

$$
\begin{aligned}
\text{Min} \quad & z = c_1x_1 + c_2x_2 + c_3x_3 && (0.1)\\
\text{s.t.} \quad & a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \geq b_1 && (0.2)\\
& a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \leq b_2 && (0.3)\\
& a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 && (0.4)\\
& a_{41}x_1 + a_{42}x_2 + a_{43}x_3 \geq b_4 && (0.5)\\
& x_1 \geq 0, x_2 \leq 0, x_3 \ urs. && (0.6)
\end{aligned}
$$

Eq. (**??**) is the objective function, (**??**)-(**??**) are the functional constraints, while (**??**) is the sign restrictions (urs signifies that the variable is unrestricted). If we were to add any one of these following constraints $x_2 \in \{0, 1\}$ (x_2 is binary-valued) or $x_3 \in \mathscr{Z}$ (x_3 is integer-valued) we would have an Integer Program. For the purposes of this class, an Integer Program (IP) is just an LP with added integer restrictions on (some) variables.

While, in general, solvers will take any form of the LP, there are some special forms we use in analysis:

LP Standard Form: The standard form has all constraints as equalities, and all variables as non-negative. The generic LP is not in standard form, but any LP can be converted to standard form.

Since x_2 is non-positive and x_3 unrestricted, perform the following substitutions $x_2 = -\hat{x}_2$ and $x_3 = x_3^+ - x_3^-$, where $\hat{x}_2, x_3^+, x_3^- \geq 0$. Eqs. (**??**) and (**??**) are in the form left-hand side (LHS) $\geq$ right-hand side (RHS), so to make an equality, subtract a non-negative slack variable from the LHS (s_1 and s_4). Eq. (**??**) is in the form LHS $\leq$ RHS, so add a non-negative slack variable to the LHS.

$$\begin{aligned}
\text{Min} \quad & z = c_1x_1 - c_2\hat{x}_2 + c_3(x_3^+ - x_3^-) \\
\text{s.t.} \quad & a_{11}x_1 - a_{12}x_2 + a_{13}(x_3^+ - x_3^-) - s_1 = b_1 \\
& a_{21}x_1 - a_{22}\hat{x}_2 + a_{23}(x_3^+ - x_3^-) + s_2 = b_2 \\
& a_{31}x_1 - a_{32}\hat{x}_2 + a_{33}(x_3^+ - x_3^-) = b_3 \\
& a_{41}x_1 - a_{42}\hat{x}_2 + a_{43}x_3 - s_4 = b_4 \\
& x_1, \hat{x}_2, x_3^+, x_3^-, s_1, s_2, s_4 \geq 0.
\end{aligned}$$

LP Canonical Form: For a minimization problem the canonical form of the LP has the LHS of each constraint greater than or equal to the the RHS, and a maximization the LHS less than or equal to the RHS, and non-negative variables.

Next we consider some formulation examples:

Production Problem: You have 21 units of transparent aluminum alloy (TAA), LazWeld1, a joining robot leased for 23 hours, and CrumCut1, a cutting robot leased for 17 hours of aluminum cutting. You also have production code for a bookcase, desk, and cabinet, along with commitments to buy any of these you can produce for \$18, \$16, and \$10 apiece, respectively. A bookcase requires 2 units of TAA, 3 hours of joining, and 1 hour of cutting, a desk requires 2 units of TAA, 2 hours of joining, and 2 hour of cutting, and a cabinet requires 1 unit of TAA, 2 hours of joining, and 1 hour of cutting. Formulate an LP to maximize your revenue given your current resources.

<u>Decision variables:</u>
x_i : number of units of product i to produce,
$\forall i = \{bookcase,\ desk,\ cabinet\}$.

$$\begin{aligned}
\max z = 18x_1 + 16x_2 + 10x_3 : & \\
2x_1 + 2x_2 + 1x_3 \leq 21 & \qquad (TAA) \\
3x_1 + 2x_2 + 2x_3 \leq 23 & \qquad (LazWeld1) \\
1x_1 + 2x_2 + 1x_3 \leq 17 & \qquad (CrumCut1) \\
x_1, x_2, x_3 \geq 0. &
\end{aligned}$$

Work Scheduling Problem: You are the manager of LP Burger. The following table shows the minimum number of employees required to staff the restaurant on each day of the week. Each employees must work

Day of Week	Workers Required
1 = Monday	6
2 = Tuesday	4
3 = Wednesday	5
4 = Thursday	4
5 = Friday	3
6 = Saturday	7
7 = Sunday	7

for five consecutive days. Formulate an LP to find the minimum number of employees required to staff the restaurant.

<u>Decision variables:</u>
x_i : the number of workers that start 5 consecutive days of work on day i, $i = 1, \cdots, 7$

$$
\begin{aligned}
\text{Min} \quad & z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \\
\text{s.t.} \quad & x_1 + x_4 + x_5 + x_6 + x_7 \geq 6 \\
& x_2 + x_5 + x_6 + x_7 + x_1 \geq 4 \\
& x_3 + x_6 + x_7 + x_1 + x_2 \geq 5 \\
& x_4 + x_7 + x_1 + x_2 + x_3 \geq 4 \\
& x_5 + x_1 + x_2 + x_3 + x_4 \geq 3 \\
& x_6 + x_2 + x_3 + x_4 + x_5 \geq 7 \\
& x_7 + x_3 + x_4 + x_5 + x_6 \geq 7 \\
& x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0.
\end{aligned}
$$

The solution is as follows:

LP Solution	IP Solution
$z_{LP} = 7.333$	$z_I = 8.0$
$x_1 = 0$	$x_1 = 0$
$x_2 = 0.333$	$x_2 = 0$
$x_3 = 1$	$x_3 = 0$
$x_4 = 2.333$	$x_4 = 3$
$x_5 = 0$	$x_5 = 0$
$x_6 = 3.333$	$x_6 = 4$
$x_7 = 0.333$	$x_7 = 1$

LP Burger has changed it's policy, and allows, at most, two part time workers, who work for two consecutive days in a week. Formulate this problem.

Decision variables:
x_i : the number of workers that start 5 consecutive days of work on day i, $i = 1, \cdots, 7$
y_i : the number of workers that start 2 consecutive days of work on day i, $i = 1, \cdots, 7$.

$$
\begin{aligned}
\text{Min} \quad & z = 5(x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7) \\
& + 2(y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7) \\
\text{s.t.} \quad & x_1 + x_4 + x_5 + x_6 + x_7 + y_1 + y_7 \geq 6 \\
& x_2 + x_5 + x_6 + x_7 + x_1 + y_2 + y_1 \geq 4 \\
& x_3 + x_6 + x_7 + x_1 + x_2 + y_3 + y_2 \geq 5 \\
& x_4 + x_7 + x_1 + x_2 + x_3 + y_4 + y_3 \geq 4 \\
& x_5 + x_1 + x_2 + x_3 + x_4 + y_5 + y_4 \geq 3 \\
& x_6 + x_2 + x_3 + x_4 + x_5 + y_6 + y_5 \geq 7 \\
& x_7 + x_3 + x_4 + x_5 + x_6 + y_7 + y_6 \geq 7 \\
& y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 \leq 2 \\
& x_i \geq 0, y_i \geq 0, \forall i = 1, \cdots, 7.
\end{aligned}
$$

The Diet Problem: In the future (as envisioned in a bad 70's science fiction film) all food is in tablet form, and there are four types, green, blue, yellow, and red. A balanced, futuristic diet requires, at least 20 units of Iron, 25 units of Vitamin B, 30 units of Vitamin C, and 15 units of Vitamin D. Formulate an LP that ensures a balanced diet at the minimum possible cost.

Tablet	Iron	B	C	D	Cost ($)
green (1)	6	6	7	4	1.25
blue (2)	4	5	4	9	1.05
yellow (3)	5	2	5	6	0.85
red (4)	3	6	3	2	0.65

Now we formulate the problem:

Decision variables:
x_i : number of tablet of type i to include in the diet, $\forall i \in \{1, 2, 3, 4\}$.

$$
\begin{aligned}
\text{Min} \quad & z = 1.25x_1 + 1.05x_2 + 0.85x_3 + 0.65x_4 \\
\text{s.t.} \quad & 6x_1 + 4x_2 + 5x_3 + 3x_4 \geq 20 \\
& 6x_1 + 5x_2 + 2x_3 + 6x_4 \geq 25 \\
& 7x_1 + 4x_2 + 5x_3 + 3x_4 \geq 30 \\
& 4x_1 + 9x_2 + 6x_3 + 2x_4 \geq 15 \\
& x_1, x_2, x_3, x_4 \geq 0.
\end{aligned}
$$

The Next Diet Problem: Progress is important, and our last problem had too many tablets, so we are going to produce a single, purple, 10 gram tablet for our futuristic diet requires, which are at least 20 units of Iron, 25 units of Vitamin B, 30 units of Vitamin C, and 15 units of Vitamin D, and 2000 calories. The tablet is made from blending 4 nutritious chemicals; the following table shows the units of our nutrients per, and cost of, grams of each chemical. Formulate an LP that ensures a balanced diet at the minimum

Tablet	Iron	B	C	D	Calories	Cost ($)
Chem 1	6	6	7	4	1000	1.25
Chem 2	4	5	4	9	250	1.05
Chem 3	5	2	5	6	850	0.85
Chem 4	3	6	3	2	750	0.65

possible cost.

<u>Decision variables:</u>
x_i : grams of chemical i to include in the purple tablet, $\forall i = 1,2,3,4$.

$$
\begin{aligned}
\text{Min} z &= 1.25x_1 + 1.05x_2 + 0.85x_3 + 0.65x_4 \\
\text{s.t. } & 6x_1 + 4x_2 + 5x_3 + 3x_4 \geq 20 \\
& 6x_1 + 5x_2 + 2x_3 + 6x_4 \geq 25 \\
& 7x_1 + 4x_2 + 5x_3 + 3x_4 \geq 30 \\
& 4x_1 + 9x_2 + 6x_3 + 2x_4 \geq 15 \\
& 1000x_1 + 250x_2 + 850x_3 + 750x_4 \geq 2000 \\
& x_1 + x_2 + x_3 + x_4 = 10 \\
& x_1, x_2, x_3, x_4 \geq 0.
\end{aligned}
$$

The Assignment Problem: Consider the assignment of n teams to n projects, where each team ranks the projects, where their favorite project is given a rank of n, their next favorite $n-1$, and their least favorite project is given a rank of 1. The assignment problem is formulated as follows (we denote ranks using the R-parameter):

<u>Variables:</u>
x_{ij} : 1 if project i assigned to team j, else 0.

$$
\begin{aligned}
\text{Max } z &= \sum_{i=1}^{n} \sum_{j=1}^{n} R_{ij} x_{ij} \\
\text{s.t. } & \sum_{i=1}^{n} x_{ij} = 1, \ \ \forall j = 1, \cdots, n \\
& \sum_{j=1}^{n} x_{ij} = 1, \ \ \forall i = 1, \cdots, n \\
& x_{ij} \geq 0, \ \ \forall i = 1, \cdots, n, j = 1, \cdots, n.
\end{aligned}
$$

The assignment problem has an integrality property, such that if we remove the binary restriction on the x variables (now just non-negative, i.e., $x_{ij} \geq 0$) then we still get binary assignments, despite the fact that it is now an LP. This property is very interesting and useful. Of course, the objective function might not quite what we want, we might be interested ensuring that the team with the worst assignment is as good as possible (a fairness criteria). One way of doing this is to modify the assignment problem using a max-min objective:

Max-min Assignment-like Formulation

$$
\begin{aligned}
Max \quad & z \\
s.t. \quad & \sum_{i=1}^{n} x_{ij} = 1, \;\; \forall j = 1, \cdots, n \\
& \sum_{j=1}^{n} x_{ij} = 1, \;\; \forall i = 1, \cdots, n \\
& x_{ij} \geq 0, \;\; \forall i = 1, \cdots, n, J = 1, \cdots, n \\
& z \leq \sum_{i=1}^{n} R_{ij} x_{ij}, \;\; \forall j = 1, \cdots, n.
\end{aligned}
$$

Does this formulation have the integrality property (it is not an assignment problem)? Consider a very simple example where two teams are to be assigned to two projects and the teams give the projects the following rankings: Both teams prefer Project 2. For both problems, if we remove the binary restriction on

	Project 1	Project 2
Team 1	2	1
Team 2	2	1

the x-variable, they can take values between (and including) zero and one. For the assignment problem the optimal solution will have $z = 3$, and fractional x-values will not improve z. For the max-min assignment problem this is not the case, the optimal solution will have $z = 1.5$, which occurs when each team is assigned half of each project (i.e., for Team 1 we have $x_{11} = 0.5$ and $x_{21} = 0.5$).

Linear Data Models: Consider a data set that consists of n data points (x_i, y_i). We want to fit the best line to this data, such that given an x-value, we can predict the associated y-value. Thus, the form is $y_i = \alpha x_i + \beta$ and we want to choose the α and β values such that we minimize the error for our n data points.

<u>Variables:</u>
e_i : error for data point i, $i = 1, \cdots, n$.
α : slope of fitted line.
β : intercept of fitted line.

$$\begin{aligned} Min \quad & \sum_{i=1}^{n} |e_i| \\ s.t. \quad & \alpha x_i + \beta - y_i = e_i, \;\; i = 1, \cdots, n \\ & e_i, \alpha, \beta \; urs. \end{aligned}$$

Of course, absolute values are not linear function, so we can linearize as follows:

<u>Decision variables:</u>
e_i^+ : positive error for data point i, $i = 1, \cdots, n$.
e_i^- : negative error for data point i, $i = 1, \cdots, n$.
α : slope of fitted line.
β : intercept of fitted line.

$$\begin{aligned} Min \quad & \sum_{i=1}^{n} e_i^+ + e_i^- \\ s.t. \quad & \alpha x_i + \beta - y_i = e_i^+ - e_i^-, \;\; i = 1, \cdots, n \\ & e_i^+, e_i^- \geq 0, \alpha, \beta \; urs. \end{aligned}$$

Two-Person Zero-Sum Games: Consider a game with two players, $\mathscr{A}$ and $\mathscr{B}$. In each round of the game, $\mathscr{A}$ chooses one out of m possible actions, while $\mathscr{B}$ chooses one out of n actions. If $\mathscr{A}$ takes action j while $\mathscr{B}$ takes action i, then c_{ij} is the payoff for $\mathscr{A}$, if $c_{ij} > 0$, $\mathscr{A}$ “wins" c_{ij} (and $\mathscr{B}$ losses that amount), and if $c_{ij} < 0$ if $\mathscr{B}$ “wins" $-c_{ij}$ (and $\mathscr{A}$ losses that amount). This is a two-person zero-sum game.

Rock, Paper, Scissors is a two-person zero-sum game, with the following payoff matrix.

$\mathscr{A}$

		R	**P**	**S**
	R	0	1	-1
$\mathscr{B}$	**P**	-1	0	1
	S	1	-1	0

We can have a similar game, but with a different payoff matrix, as follows:

		$\mathscr{A}$		
		R	**P**	**S**
	R	4	-1	-1
$\mathscr{B}$	**P**	-2	4	-2
	S	-3	-3	4

What is the optimal strategy for $\mathscr{A}$ (for either game)? We define x_j as the probability that $\mathscr{A}$ takes action j (related to the columns). Then the payoff for $\mathscr{A}$, if $\mathscr{B}$ takes action i is $\sum_{j=1}^{m} c_{ij}x_j$. Of course, $\mathscr{A}$ does not know what action $\mathscr{B}$ will take, so let's find a strategy that maximizes the minimum expected winnings of $\mathscr{A}$ given any random strategy of $\mathscr{B}$, which we can formulate as follows:

$$\begin{aligned} Max \quad & (min_{i=1,\cdots,n} \sum_{j=1}^{m} c_{ij}x_i) \\ s.t. \quad & \sum_{j=1}^{m} x_j = 1 \\ & x_j \geq 0, \;\; i = 1,\cdots,m, \end{aligned}$$

which can be linearized as follows:

$$\begin{aligned} Max \quad & z \\ s.t. \quad & z \leq \sum_{j=1}^{m} c_{ij}x_j, \;\; i = 1,\cdots,n \\ & \sum_{j=1}^{m} x_j = 1 \\ & x_j \geq 0, \;\; i = 1,\cdots,m. \end{aligned}$$

The last two constraints ensure the that x_i-variables are valid probabilities. If you solved this LP for the first game (i.e., payoff matrix) you find the best strategy is $x_1 = 1/3$, $x_2 = 1/3$, and $x_3 = 1/3$ and there is no expected gain for player $\mathscr{A}$. For the second game, the best strategy is $x_1 = 23/107$, $x_2 = 37/107$, and $x_3 = 47/107$, with $\mathscr{A}$ gaining, on average, $8/107$ per round.

Part I

OLD LP Stuff

1. LP Notes from Foundations of Applied Mathematics

Linear Programs

A *linear program* is a linear constrained optimization problem. Such a problem can be stated in several different forms, one of which is

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^\mathsf{T}\mathbf{x} \\ \text{subject to} \quad & G\mathbf{x} \preceq \mathbf{h} \\ & A\mathbf{x} = \mathbf{b}. \end{aligned}$$

The symbol $\preceq$ denotes that the components of $G\mathbf{x}$ are less than the components of $\mathbf{h}$. In other words, if $\mathbf{x} \preceq \mathbf{y}$, then $x_i < y_i$ for all $x_i \in \mathbf{x}$ and $y_i \in \mathbf{y}$.

Define vector $\mathbf{s} \succeq 0$ such that the constraint $G\mathbf{x} + \mathbf{s} = \mathbf{h}$. This vector is known as a *slack variable*. Since $\mathbf{s} \succeq 0$, the constraint $G\mathbf{x} + \mathbf{s} = \mathbf{h}$ is equivalent to $G\mathbf{x} \preceq \mathbf{h}$.

With a slack variable, a new form of the linear program is found:

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^\mathsf{T}\mathbf{x} \\ \text{subject to} \quad & G\mathbf{x} + \mathbf{s} = \mathbf{h} \\ & A\mathbf{x} = \mathbf{b} \\ & \mathbf{s} \succeq 0. \end{aligned}$$

This is the formulation used by CVXOPT. It requires that the matrix A has full row rank, and that the block matrix $[G \quad A]^\mathsf{T}$ has full column rank.

Consider the following example:

$$\begin{aligned} \text{minimize} \quad & -4x_1 - 5x_2 \\ \text{subject to} \quad & x_1 + 2x_2 \leq 3 \\ & 2x_1 + x_2 = 3 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Recall that all inequalities must be less than or equal to, so that $G\mathbf{x} \leq \mathbf{h}$. Because the final two constraints are $x_1, x_2 \geq 0$, they need to be adjusted to be $\leq$ constraints. This is easily done by multiplying by -1, resulting in the constraints $-x_1, -x_2 \leq 0$. If we define

$$G = \begin{bmatrix} 1 & 2 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad \mathbf{h} = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}, \qquad A = \begin{bmatrix} 2 & 1 \end{bmatrix}, \quad \text{and} \qquad \mathbf{b} = \begin{bmatrix} 3 \end{bmatrix}$$

then we can express the constraints compactly as

$$\begin{array}{l} G\mathbf{x} \leq \mathbf{h}, \\ A\mathbf{x} = \mathbf{b}, \end{array} \qquad \text{where} \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

By adding a slack variable $\mathbf{s}$, we can write our constraints as

$$G\mathbf{x} + \mathbf{s} = \mathbf{h},$$

which matches the form discussed above.

Problem 1.1: Linear Optimization

Solve the following linear optimization problem:

$$\begin{array}{rl} \textit{minimize} & 2x_1 + x_2 + 3x_3 \\ \textit{subject to} & x_1 + 2x_2 \geq 3 \\ & 2x_1 + 10x_2 + 3x_3 \geq 10 \\ & x_i \geq 0 \ \ \textit{for}\ i = 1,2,3 \end{array}$$

Return the minimizer $\mathbf{x}$ *and the primal objective value.*
(Hint: make the necessary adjustments so that all inequality constraints are $\leq$ *rather than* $\geq$*).*

l_1 Norm

The l_1 norm is defined

$$||\mathbf{x}||_1 = \sum_{i=1}^{n} |x_i|.$$

A l_1 minimization problem is minimizing a vector's l_1 norm, while fitting certain constraints. It can be written in the following form:

$$\begin{array}{rl} \text{minimize} & \|\mathbf{x}\|_1 \\ \text{subject to} & A\mathbf{x} = \mathbf{b}. \end{array}$$

This problem can be converted into a linear program by introducing an additional vector $\mathbf{u}$ of length n. Define $\mathbf{u}$ such that $|x_i| \leq u_i$. Thus, $-u_i - x_i \leq 0$ and $-u_i + x_i \leq 0$. These two inequalities can be added to the linear system as constraints. Additionally, this means that $||\mathbf{x}||_1 \leq ||\mathbf{u}||_1$. So minimizing $||\mathbf{u}||_1$ subject to the given constraints will in turn minimize $||\mathbf{x}||_1$. This can be written as follows:

$$\begin{aligned} \text{minimize} \quad & \begin{bmatrix} \mathbb{1}^\mathsf{T} & 0^\mathsf{T} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{x} \end{bmatrix} \\ \text{subject to} \quad & \begin{bmatrix} -I & I \\ -I & -I \\ -I & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{x} \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ & \begin{bmatrix} 0 & A \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{x} \end{bmatrix} = \mathbf{b}. \end{aligned}$$

Solving this gives values for the optimal $\mathbf{u}$ and the optimal $\mathbf{x}$, but we only care about the optimal $\mathbf{x}$.

Problem 1.2: ℓ_1 Norm Minimization

rite a function called `l1Min()` that accepts a matrix A and vector $\mathbf{b}$ as NumPy arrays and solves the l_1 minimization problem. Return the minimizer $\mathbf{x}$ and the primal objective value. Remember to first discard the unnecessary u values from the minimizer.
To test your function consider the matrix A and vector $\mathbf{b}$ below.

$$A = \begin{bmatrix} 1 & 2 & 1 & 1 \\ 0 & 3 & -2 & -1 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 7 \\ 4 \end{bmatrix}$$

The linear system $A\mathbf{x} = \mathbf{b}$ has infinitely many solutions. Use `l1Min()` to verify that the solution which minimizes $||\mathbf{x}||_1$ is approximately $\mathbf{x} = [0., 2.571, 1.857, 0.]^T$ and the minimum objective value is approximately 4.429.

The Transportation Problem

Consider the following transportation problem: A piano company needs to transport thirteen pianos from their three supply centers (denoted by 1, 2, 3) to two demand centers (4, 5). Transporting a piano from a supply center to a demand center incurs a cost, listed in Table **??**. The company wants to minimize shipping costs for the pianos while meeting the demand.

Supply Center	Number of pianos available
1	7
2	2
3	4

Table 1.1: Number of pianos available at each supply center

Demand Center	Number of pianos needed
4	5
5	8

Table 1.2: Number of pianos needed at each demand center

Supply Center	Demand Center	Cost of transportation	Number of pianos
1	4	4	p_1
1	5	7	p_2
2	4	6	p_3
2	5	8	p_4
3	4	8	p_5
3	5	9	p_6

Table 1.3: Cost of transporting one piano from a supply center to a demand center

A system of constraints is defined for the variables p_1, p_2, p_3, p_4, p_5, and p_6, First, there cannot be a negative number of pianos so the variables must be nonnegative. Next, the Tables **??** and **??** define the following three supply constraints and two demand constraints:

$$\begin{aligned} p_1 + p_2 &= 7 \\ p_3 + p_4 &= 2 \\ p_5 + p_6 &= 4 \\ p_1 + p_3 + p_5 &= 5 \\ p_2 + p_4 + p_6 &= 8 \end{aligned}$$

The objective function is the number of pianos shipped from each location multiplied by the respective cost (found in Table **??**):

$$4p_1 + 7p_2 + 6p_3 + 8p_4 + 8p_5 + 9p_6.$$

Note

Since our answers must be integers, in general this problem turns out to be an NP-hard problem. There is a whole field devoted to dealing with integer constraints, called *integer linear programming*, which is beyond the scope of this lab. Fortunately, we can treat this particular problem as a standard linear program and still obtain integer solutions.

Recall the variables are nonnegative, so $p_1, p_2, p_3, p_4, p_5, p_6 \geq 0$. Thus, G and $\mathbf{h}$ constrain the variables to be non-negative.

Problem 1.3: Transportation problem

Solve the transportation problem by converting the last equality constraint into an inequality constraint. Return the minimizer $\mathbf{x}$ *and the primal objective value.*

Eating on a Budget

In 2009, the inmates of Morgan County jail convinced Judge Clemon of the Federal District Court in Birmingham to put Sheriff Barlett in jail for malnutrition. Under Alabama law, in order to encourage less spending, "the chief lawman could go light on prisoners' meals and pocket the leftover change."[1]. Sheriffs had to ensure a minimum amount of nutrition for inmates, but minimizing costs meant more money for the sheriffs themselves. Judge Clemon jailed Sheriff Barlett one night until a plan was made to use all allotted funds, 1.75 per inmate, to feed prisoners more nutritious meals. While this case made national news, the controversy of feeding prisoners in Alabama continues as of 2019[2].

The problem of minimizing cost while reaching healthy nutritional requirements can be approached as a convex optimization problem. Rather than viewing this problem from the sheriff's perspective, we view it from the perspective of a college student trying to minimize food cost in order to pay for higher education, all while meeting standard nutritional guidelines.

The file `food.npy` contains a dataset with nutritional facts for 18 foods that have been eaten frequently by college students working on this text. A subset of this dataset can be found in Table **??**, where the "Food" column contains the list of all 18 foods.

The columns of the full dataset are:

Column 1: p, price (dollars)
Column 2: s, number of servings
Column 3: c, calories per serving
Column 4: f, fat per serving (grams)
Column 5: $\hat{s}$, sugar per serving (grams)
Column 6: $\hat{c}$, calcium per serving (milligrams)
Column 7: $\hat{f}$, fiber per serving (grams)
Column 8: $\hat{p}$, protein per serving (grams)

[1]Nossiter, Adam, 8 Jan 2009, "As His Inmates Grew Thinner, a Sheriff's Wallet Grew Fatter", *New York Times*, `https://www.nytimes.com/2009/01/09/us/09sheriff.html`

[2]Sheets, Connor, 31 January 2019, "Alabama sheriffs urge lawmakers to get them out of the jail food business", `https://www.al.com/news/2019/01/alabama-sheriffs-urge-lawmakers-to-get-them-out-of-the-jail-food-business.html`

Food	Price p dollars	Serving Size s	Calories c	Fat f g	Sugar $\hat{s}$ g	Calcium $\hat{c}$ mg	Fiber $\hat{f}$ g	Protein $\hat{p}$ g
Ramen	6.88	48	190	7	0	0	0	5
Potatoes	0.48	1	290	0.4	3.2	53.8	6.9	7.9
Milk	1.79	16	130	5	12	250	0	8
Eggs	1.32	12	70	5	0	28	0	6
Pasta	3.88	8	200	1	2	0	2	7
Frozen Pizza	2.78	5	350	11	5	150	2	14
Potato Chips	2.12	14	160	11	1	0	1	1
Frozen Broccoli	0.98	4	25	0	1	25	2	1
Carrots	0.98	2	52.5	0.3	6.1	42.2	3.6	1.2
Bananas	0.24	1	105	0.4	14.4	5.9	3.1	1.3
Tortillas	3.48	18	140	4	0	0	0	3
Cheese	1.88	8	110	8	0	191	0	6
Yogurt	3.47	5	90	0	7	190	0	17
Bread	1.28	6	120	2	2	60	0.01	4
Chicken	9.76	20	110	3	0	0	0	20
Rice	8.43	40	205	0.4	0.1	15.8	0.6	4.2
Pasta Sauce	3.57	15	60	1.5	7	20	2	2
Lettuce	1.78	6	8	0.1	0.6	15.5	1	0.6

Table 1.4: Subset of table containing food data

According to the FDA[1] and US Department of Health, someone on a 2000 calorie diet should have no more than 2000 calories, no more than 65 grams of fat, no more than 50 grams of sugar[2], at least 1000 milligrams of calcium[1], at least 25 grams of fiber, and at least 46 grams of protein[2] per day.

We can rewrite this as a linear programming problem below.

[1]urlhttps://www.accessdata.fda.gov/scripts/InteractiveNutritionFactsLabel/pdv.html

[2]https://www.today.com/health/4-rules-added-sugars-how-calculate-your-daily-limit-t34731

[1]26 Sept 2018, `https://ods.od.nih.gov/factsheets/Calcium-HealthProfessional/`

[2]`https://www.accessdata.fda.gov/scripts/InteractiveNutritionFactsLabel/protein.html`

$$
\begin{aligned}
\text{minimize } & \sum_{i=1}^{18} p_i x_i, \\
\text{subject to } & \sum_{i=1}^{18} c_i x_i \leq 2000, \\
& \sum_{i=1}^{18} f_i x_i \leq 65, \\
& \sum_{i=1}^{18} \hat{s}_i x_i \leq 50, \\
& \sum_{i=1}^{18} \hat{c}_i x_i \geq 1000, \\
& \sum_{i=1}^{18} \hat{f}_i x_i \geq 25, \\
& \sum_{i=1}^{18} \hat{p}_i x_i \geq 46, \\
& x_i \geq 0.
\end{aligned}
$$

Problem 1.4: Eating on a Budget

Read in the file `food.npy`*. Identify how much of each food item a college student should each to minimize cost spent each day. Return the minimizing vector and the total amount of money spent. What is the food you should eat most each day? What are the three foods you should eat most each week?*
(Hint: Each nutritional value must be multiplied by the number of servings to get the nutrition value of the whole product).

1.1 The Simplex Method

The Simplex Method **Lab Objective:** *The* Simplex Method *is a straightforward algorithm for finding optimal solutions to optimization problems with linear constraints and cost functions. Because of its simplicity and applicability, this algorithm has been named one of the most important algorithms invented within the last 100 years. In this lab we implement a standard Simplex solver for the primal problem.*

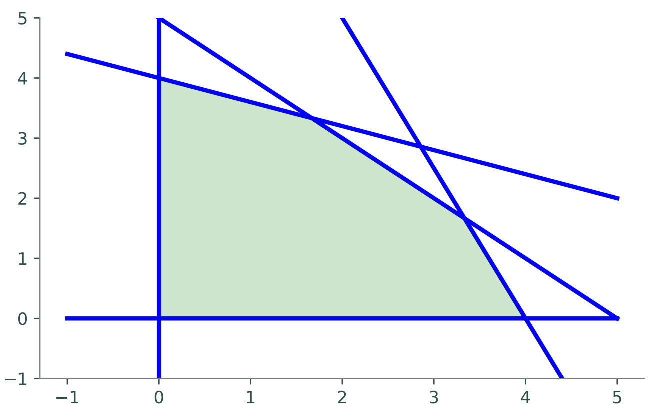
Standard Form

The Simplex Algorithm accepts a linear constrained optimization problem, also called a *linear program*, in the form given below:

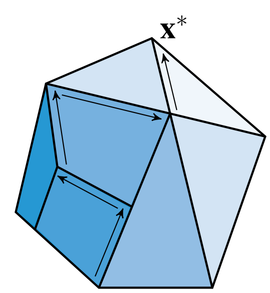
$$
\begin{aligned}
\text{minimize} \quad & \mathbf{c}^\mathsf{T}\mathbf{x} \\
\text{subject to} \quad & A\mathbf{x} \leq \mathbf{b} \\
& \mathbf{x} \geq 0
\end{aligned}
$$

Note that any linear program can be converted to standard form, so there is no loss of generality in restricting our attention to this particular formulation.

Such an optimization problem defines a region in space called the *feasible region*, the set of points satisfying the constraints. Because the constraints are all linear, the feasible region forms a geometric object called a *polytope*, having flat faces and edges (see Figure **??**). The Simplex Algorithm jumps among the vertices of the feasible region searching for an optimal point. It does this by moving along the edges of the feasible region in such a way that the objective function is always increased after each move.



(a) The feasible region for a linear program with 2-dimensional constraints.



(b) The feasible region for a linear program with 3-dimensional constraints.

Figure 1.1: If an optimal point exists, it is one of the vertices of the polyhedron. The simplex algorithm searches for optimal points by moving between adjacent vertices in a direction that increases the value of the objective function until it finds an optimal vertex.

Implementing the Simplex Algorithm is straightforward, provided one carefully follows the procedure. We will break the algorithm into several small steps, and write a function to perform each one. To become familiar with the execution of the Simplex algorithm, it is helpful to work several examples by hand.

The Simplex Solver

Our program will be more lengthy than many other lab exercises and will consist of a collection of functions working together to produce a final result. It is important to clearly define the task of each function and how all the functions will work together. If this program is written haphazardly, it will be much longer and more difficult to read than it needs to be. We will walk you through the steps of implementing the Simplex Algorithm as a Python class.

For demonstration purposes, we will use the following linear program.

$$\begin{aligned} \text{minimize} \qquad & -3x_0 - 2x_1 \\ \text{subject to} \qquad & x_0 - x_1 \leq 2 \\ & 3x_0 + x_1 \leq 5 \\ & 4x_0 + 3x_1 \leq 7 \\ & x_0, x_1 \geq 0. \end{aligned}$$

Accepting a Linear Program

Our first task is to determine if we can even use the Simplex algorithm. Assuming that the problem is presented to us in standard form, we need to check that the feasible region includes the origin. For now, we only check for feasibility at the origin. A more robust solver sets up the auxiliary problem and solves it to find a starting point if the origin is infeasible.

Problem 1.5: Check feasibility at the origin.

Write a class that accepts the arrays $\mathbf{c}$, A, *and* $\mathbf{b}$ *of a linear optimization problem in standard form. In the constructor, check that the system is feasible at the origin. That is, check that* $A\mathbf{x} \preceq \mathbf{b}$ *when* $\mathbf{x} = 0$. *Raise a* `ValueError` *if the problem is not feasible at the origin.*

Adding Slack Variables

The next step is to convert the inequality constraints $A\mathbf{x} \leq \mathbf{b}$ into equality constraints by introducing a slack variable for each constraint equation. If the constraint matrix A is an $m \times n$ matrix, then there are m slack variables, one for each row of A. Grouping all of the slack variables into a vector $\mathbf{w}$ of length m, the constraints now take the form $A\mathbf{x} + \mathbf{w} = \mathbf{b}$. In our example, we have

$$\mathbf{w} = \begin{bmatrix} x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

When adding slack variables, it is useful to represent all of your variables, both the original primal variables and the additional slack variables, in a convenient manner. One effective way is to refer to a variable by its subscript. For example, we can use the integers 0 through $n-1$ to refer to the original (non-slack) variables x_0 through x_{n-1}, and we can use the integers n through $n+m-1$ to track the slack variables (where the slack variable corresponding to the ith row of the constraint matrix is represented by the index $n+i-1$).

We also need some way to track which variables are *independent* (non-zero) and which variables are *dependent* (those that have value 0). This can be done using the objective function. At anytime during the optimization process, the non-zero variables in the objective function are *independent* and all other variables are *dependent*.

Creating a Dictionary

After we have determined that our program is feasible, we need to create the *dictionary* (sometimes called the *tableau*), a matrix to track the state of the algorithm.

There are many different ways to build your dictionary. One way is to mimic the dictionary that is often used when performing the Simplex Algorithm by hand. To do this we will set the corresponding dependent variable equations to 0. For example, if x_5 were a dependent variable we would expect to see a -1 in the column that represents x_5. Define

$$\bar{A} = \begin{bmatrix} A & I_m \end{bmatrix},$$

where I_m is the $m \times m$ identity matrix we will use to represent our slack variables, and define

$$\bar{\mathbf{c}} = \begin{bmatrix} \mathbf{c} \\ 0 \end{bmatrix}.$$

That is, $\bar{\mathbf{c}} \in \mathbb{R}^{n+m}$ such that the first n entries are $\mathbf{c}$ and the final m entries are zeros. Then the initial dictionary has the form

$$D = \begin{bmatrix} 0 & \bar{\mathbf{c}}^\mathsf{T} \\ \mathbf{b} & -\bar{A} \end{bmatrix} \tag{1.1}$$

The columns of the dictionary correspond to each of the variables (both primal and slack), and the rows of the dictionary correspond to the dependent variables.

For our example the initial dictionary is

$$D = \begin{bmatrix} 0 & -3 & -2 & 0 & 0 & 0 \\ 2 & -1 & 1 & -1 & 0 & 0 \\ 5 & -3 & -1 & 0 & -1 & 0 \\ 7 & -4 & -3 & 0 & 0 & -1 \end{bmatrix}.$$

The advantage of using this kind of dictionary is that it is easy to check the progress of your algorithm by hand.

Problem 1.6: Initialize the dictionary.

dd a method to your Simplex solver that takes in arrays c, A, and b to create the initial dictionary (D) as a NumPy array.

1.1.1. Pivoting

Pivoting is the mechanism that really makes Simplex useful. Pivoting refers to the act of swapping dependent and independent variables, and transforming the dictionary appropriately. This has the effect of moving from one vertex of the feasible polytope to another vertex in a way that increases the value of the objective function. Depending on how you store your variables, you may need to modify a few different parts of your solver to reflect this swapping.

When initiating a pivot, you need to determine which variables will be swapped. In the dictionary representation, you first find a specific element on which to pivot, and the row and column that contain the pivot element correspond to the variables that need to be swapped. Row operations are then performed on the dictionary so that the pivot column becomes a negative elementary vector.

Let's break it down, starting with the pivot selection. We need to use some care when choosing the pivot element. To find the pivot column, search from left to right along the top row of the dictionary (ignoring the first column), and stop once you encounter the first negative value. The index corresponding to this column will be designated the *entering index*, since after the full pivot operation, it will enter the basis and become a dependent variable.

Using our initial dictionary D in the example, we stop at the second column:

$$D = \left[\begin{array}{c|c|ccccc} 0 & -3 & -2 & 0 & 0 & 0 \\ 2 & -1 & 1 & -1 & 0 & 0 \\ 5 & -3 & -1 & 0 & -1 & 0 \\ 7 & -4 & -3 & 0 & 0 & -1 \end{array}\right]$$

We now know that our pivot element will be found in the second column. The entering index is thus 1.

Next, we select the pivot element from among the negative entries in the pivot column (ignoring the entry in the first row). *If all entries in the pivot column are non-negative, the problem is unbounded and has no solution.* In this case, the algorithm should terminate. Otherwise, assuming our pivot column is the jth column of the dictionary and that the negative entries of this column are $D_{i_1,j}, D_{i_2,j}, \ldots, D_{i_k,j}$, we calculate the ratios

$$\frac{-D_{i_1,0}}{D_{i_1,j}}, \frac{-D_{i_2,0}}{D_{i_2,j}}, \ldots, \frac{-D_{i_k,0}}{D_{i_k,j}},$$

and we choose our pivot element to be one that minimizes this ratio. If multiple entries minimize the ratio, then we utilize *Bland's Rule*, which instructs us to choose the entry in the row corresponding to the smallest index (obeying this rule is important, as it prevents the possibility of the algorithm cycling back on itself infinitely). The index corresponding to the pivot row is designated as the *leaving index*, since after the full pivot operation, it will leave the basis and become a independent variable.

In our example, we see that all entries in the pivot column (ignoring the entry in the first row, of course) are negative, and hence they are all potential choices for the pivot element. We then calculate the ratios, and obtain

$$\frac{-2}{-1} = 2, \quad \frac{-5}{-3} = 1.66..., \quad \frac{-7}{-4} = 1.75.$$

We see that the entry in the third row minimizes these ratios. Hence, the element in the second column (index 1), third row (index 2) is our designated pivot element.

$$D = \begin{bmatrix} 0 & -3 & -2 & 0 & 0 & 0 \\ 2 & -1 & 1 & -1 & 0 & 0 \\ 5 & \boxed{-3} & -1 & 0 & -1 & 0 \\ 7 & -4 & -3 & 0 & 0 & -1 \end{bmatrix}$$

Definition 1.7: Bland's Rule

hoose the independent variable with the smallest index that has a negative coefficient in the objective function as the leaving variable. Choose the dependent variable with the smallest index among all the binding dependent variables.

Bland's Rule is important in avoiding cycles when performing pivots. This rule guarantees that a feasible Simplex problem will terminate in a finite number of pivots.

Finally, we perform row operations on our dictionary in the following way: divide the pivot row by the negative value of the pivot entry. Then use the pivot row to zero out all entries in the pivot column above and below the pivot entry. In our example, we first divide the pivot row by -3, and then zero out the two entries above the pivot element and the single entry below it:

$$\begin{bmatrix} 0 & -3 & -2 & 0 & 0 & 0 \\ 2 & -1 & 1 & -1 & 0 & 0 \\ 5 & -3 & -1 & 0 & -1 & 0 \\ 7 & -4 & -3 & 0 & 0 & -1 \end{bmatrix} \to \begin{bmatrix} 0 & -3 & -2 & 0 & 0 & 0 \\ 2 & -1 & 1 & -1 & 0 & 0 \\ 5/3 & -1 & -1/3 & 0 & -1/3 & 0 \\ 7 & -4 & -3 & 0 & 0 & -1 \end{bmatrix} \to$$
$$\begin{bmatrix} -5 & 0 & -1 & 0 & 1 & 0 \\ 2 & -1 & 1 & -1 & 0 & 0 \\ 5/3 & -1 & -1/3 & 0 & -1/3 & 0 \\ 7 & -4 & -3 & 0 & 0 & -1 \end{bmatrix} \to \begin{bmatrix} -5 & 0 & -1 & 0 & 1 & 0 \\ 1/3 & 0 & -4/3 & 1 & -1/3 & 0 \\ 5/3 & -1 & -1/3 & 0 & -1/3 & 0 \\ 7 & -4 & -3 & 0 & 0 & -1 \end{bmatrix} \to$$
$$\begin{bmatrix} -5 & 0 & -1 & 0 & 1 & 0 \\ 1/3 & 0 & 4/3 & -1 & 1/3 & 0 \\ 5/3 & -1 & -1/3 & 0 & -1/3 & 0 \\ 1/3 & 0 & -5/3 & 0 & 4/3 & -1 \end{bmatrix}.$$

The result of these row operations is our updated dictionary, and the pivot operation is complete.

Problem 1.8: Pivoting

dd a method to your solver that checks for unboundedness and performs a single pivot operation from start to completion. If the problem is unbounded, raise a `ValueError`.

1.1.2. Termination and Reading the Dictionary

Up to this point, our algorithm accepts a linear program, adds slack variables, and creates the initial dictionary. After carrying out these initial steps, it then performs the pivoting operation iteratively until the optimal point is found. But how do we determine when the optimal point is found? The answer is to look at the top row of the dictionary, which represents the objective function. More specifically, before each pivoting operation, check whether all of the entries in the top row of the dictionary (ignoring the entry in the first column) are nonnegative. If this is the case, then we have found an optimal solution, and so we terminate the algorithm.

The final step is to report the solution. The ending state of the dictionary and index list tells us everything we need to know. The minimal value attained by the objective function is found in the upper leftmost entry of the dictionary. The dependent variables all have the value 0 in the objective function or first row of our dictionary array. The independent variables have values given by the first column of the dictionary. Specifically, the independent variable whose index is located at the ith entry of the index list has the value $T_{i+1,0}$.

In our example, suppose that our algorithm terminates with the dictionary and index list in the following state:

$$D = \begin{bmatrix} -5.2 & 0 & 0 & 0 & 0.2 & 0.6 \\ 0.6 & 0 & 0 & -1 & 1.4 & -0.8 \\ 1.6 & -1 & 0 & 0 & -0.6 & 0.2 \\ 0.2 & 0 & -1 & 0 & 0.8 & -0.6 \end{bmatrix}$$

Then the minimal value of the objective function is -5.2. The independent variables have indices $4,5$ and have the value 0. The dependent variables have indices $3,1$, and 2, and have values $.6,1.6$, and $.2$, respectively. In the notation of the original problem statement, the solution is given by

$$x_0 = 1.6$$
$$x_1 = .2.$$

Problem 1.9: SimplexSolver.solve()

Write an additional method in your solver called `solve()` *that obtains the optimal solution, then returns the minimal value, the dependent variables, and the independent variables. The dependent and independent variables should be represented as two dictionaries that map the index of the variable to its corresponding value.*
For our example, we would return the tuple
`(-5.2, {0: 1.6, 1: .2, 2: .6}, {3: 0, 4: 0}).`

At this point, you should have a Simplex solver that is ready to use. The following code demonstrates how your solver is expected to behave:

```python
>>> import SimplexSolver

# Initialize objective function and constraints.
>>> c = np.array([-3., -2.])
>>> b = np.array([2., 5, 7])
>>> A = np.array([[1., -1], [3, 1], [4, 3]])

# Instantiate the simplex solver, then solve the problem.
>>> solver = SimplexSolver(c, A, b)
>>> sol = solver.solve()
>>> print(sol)
(-5.2,
 {0: 1.6, 1: 0.2, 2: 0.6},
 {3: 0, 4: 0})
```

If the linear program were infeasible at the origin or unbounded, we would expect the solver to alert the user by raising an error.

Note that this simplex solver is *not* fully operational. It can't handle the case of infeasibility at the origin. This can be fixed by adding methods to your class that solve the *auxiliary problem*, that of finding an initial feasible dictionary when the problem is not feasible at the origin. Solving the auxiliary problem involves pivoting operations identical to those you have already implemented, so adding this functionality is not overly difficult.

1.2 The Product Mix Problem

We now use our Simplex implementation to solve the *product mix problem*, which in its dependent form can be expressed as a simple linear program. Suppose that a manufacturer makes n products using m different resources (labor, raw materials, machine time available, etc). The ith product is sold at a unit price p_i, and there are at most m_j units of the jth resource available. Additionally, each unit of the ith product requires $a_{j,i}$ units of resource j. Given that the demand for product i is d_i units per a certain time period, how do we choose the optimal amount of each product to manufacture in that time period so as to maximize revenue, while not exceeding the available resources?

Let $x_1, x_2, \ldots, x_n$ denote the amount of each product to be manufactured. The sale of product i brings revenue in the amount of $p_i x_i$. Therefore our objective function, the profit, is given by

$$\sum_{i=1}^{n} p_i x_i.$$

Additionally, the manufacture of product i requires $a_{j,i} x_i$ units of resource j. Thus we have the resource constraints

$$\sum_{i=1}^{n} a_{j,i} x_i \leq m_j \text{ for } j = 1, 2, \ldots, m.$$

Finally, we have the demand constraints which tell us not to exceed the demand for the products:

$$x_i \leq d_i \text{ for } i = 1, 2, \ldots, n$$

The variables x_i are constrained to be nonnegative, of course. We therefore have a linear program in the appropriate form that is feasible at the origin. It is a simple task to solve the problem using our Simplex solver.

Problem 1.10: Product mix problem.

Solve the product mix problem for the data contained in the file `productMix.npz`. *In this problem, there are 4 products and 3 resources. The archive file, which you can load using the function* `np.load`, *contains a dictionary of arrays. The array with key* `'A'` *gives the resource coefficients* $a_{i,j}$ *(i.e. the* (i,j)*-th entry of the array give* $a_{i,j}$*). The array with key* `'p'` *gives the unit prices* p_i*. The array with key* `'m'` *gives the available resource units* m_j*. The array with key* `'d'` *gives the demand constraints* d_i*.*
Report the number of units that should be produced for each product. Hint: *Because this is a maximization problem and your solver works with minimizations, you will need to change the sign of the array c.*

Beyond Simplex

The *Computing in Science and Engineering* journal listed Simplex as one of the top ten algorithms of the twentieth century [**Nash2000**]. However, like any other algorithm, Simplex has its drawbacks.

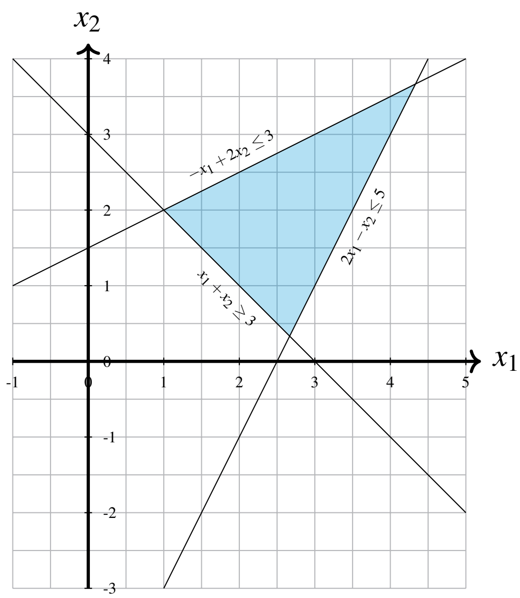
In 1972, Victor Klee and George Minty Cube published a paper with several examples of worst-case polytopes for the Simplex algorithm [**Klee1972**]. In their paper, they give several examples of polytopes that the Simplex algorithm struggles to solve.

Consider the following linear program from Klee and Minty.

$$\begin{array}{rlllll}
\max & 2^{n-1}x_1 & +2^{n-2}x_2 & +\cdots & +2x_{n-1} & +x_n \\
\text{subject to} & x_1 & & & & \leq 5 \\
& 4x_1 & +x_2 & & & \leq 25 \\
& 8x_1 & +4x_2 & +x_3 & & \leq 125 \\
& \vdots & & & & \vdots \\
& 2^n x_1 & +2^{n-1}x_2 & +\cdots & +4x_{n-1} & +x_n \leq 5
\end{array}$$

Klee and Minty show that for this example, the worst case scenario has exponential time complexity. With only n constraints and n variables, the simplex algorithm goes through 2^n iterations. This is because there are 2^n extreme points, and when starting at the point $x = 0$, the simplex algorithm goes through all of the extreme points before reaching the optimal point $(0, 0, \ldots, 0, 5^n)$. Other algorithms, such as interior point methods, solve this problem much faster because they are not constrained to follow the edges.

2. Linear Programming Notes - Hildebrand



[1]

2.0.0.1. Simplex Tableau Pivoter

`http://www.tutor-homework.com/Simplex_Tableau_Homework_Help.html`

2.0.0.2. Videos

Geometry of the simplex method: Fantastic video by Craig Torey of Georgia Tech explainging geometry of pivots and why the simplex method is called the simplx method. There is also a bit of history about Dantzig in the video.

`https://www.youtube.com/watch?v=Ci1vBGn9yRc&ab_channel=LouisHolley`

For some nice videos of doing simplex method with tableaus, I recommend:

`https://www.youtube.com/watch?v=M8POtpPtQZc`

LPP using simplex method [Minimization with 3 variables]: `https://youtu.be/SNc9NGCJmns`

LPP using Dual simplex method : `https://youtu.be/KLHWtBpPbEc`

LPP using TWO PHASE method: `https://youtu.be/zJhncZ5XUSU`

[1] `https://tex.stackexchange.com/questions/75933/how-to-draw-the-region-of-inequality`

LPP using BIG M method: `https://youtu.be/MZ843Vvia0A`

[1] LPP using Graphical method [Maximization with 2 constraints]: `https://youtu.be/8IRrgDoV8Eo`

[2] LPP using Graphical method [Minimization with 3 constraints]: `https://youtu.be/06QO3J_85as`

2.1 Linear Programming Forms

2.2 Linear Programming Dual

Consider the linear program in standard form. The dual is the following problem

Dual of LP in Standard Form:

Polynomial time (P)

Primal

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

Dual

$$\begin{aligned} \min \quad & b^\top y \\ \text{s.t.} \quad & A^\top y \geq c \\ & y \text{ free} \end{aligned} \tag{2.1}$$

2.3 Weak and Strong Duality

Theorem 2.1: Weak Duality

Let x be feasible for the primal LP and y feasible for the dual LP. Then

$$c^\top x \leq b^\top y. \tag{2.1}$$

Theorem 2.2: Strong Duality

The primal LP is feasible and has a bounded objective value if and only if the dual LP is also feasible and has a bounded objective value. In this case, the optimal values to both problems coincide. In particular, suppose x^* is optimal for the primal LP and y^* is optimal for the dual LP. Then

$$c^\top x^* = b^\top y^*. \tag{2.2}$$

2.3.1. Reduced Costs

Consider the LP in standard form (**??**) given by

$$\begin{aligned} \max \quad & c^\top x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned} \tag{2.3}$$

A basis B is a subset of the columns of A that form an invertible matrix. The remaining columns for the matrix N, that is, $A = (A_B \,|\, A_N)$ (after permuting the columns of A).

The *basic variables* x_B are the variables corresponding to the columns of A_B and the *non-basic variables* are those corresponding to the columns of A_N.

Since A_B is invertible we can convert the formulation by multiplying through by A_B^{-1}. This produces

$$\begin{aligned} \max \quad & c^\top x \\ \text{s.t.} \quad & A_B^{-1} A x = A_B^{-1} b \\ & x \geq 0 \end{aligned} \tag{2.4}$$

Since $A = (A_B \,|\, A_N)$, we have

$$\begin{aligned} \max \quad & c^\top x \\ \text{s.t.} \quad & (A_B^{-1} A_B, A_B^{-1} A_N) x = A_B^{-1} b \\ & x \geq 0 \end{aligned} \tag{2.5}$$

which becomes

$$\begin{aligned} \max \quad & c^\top x \\ \text{s.t.} \quad & x_B + A_B^{-1} A_N x_N = A_B^{-1} b \\ & x \geq 0 \end{aligned} \tag{2.6}$$

$B^{-1} b \geq 0$, then $x_b = B^{-1} b, x_N = 0$ called a *basic feasible solution*.

Manipulating the formulation again, we can multiply the equations by c_B and substract that from the objective function. This leaves us with

$$\begin{aligned} \max \quad & c_N x_N - c_B A_B^{-1} A_N x_N + c_B A_B^{-1} b \\ \text{s.t.} \quad & x_B + A_B^{-1} A_N x_N = A_B^{-1} b \\ & x \geq 0 \end{aligned} \tag{2.7}$$

combining terms creates

$$\begin{aligned} \max \quad & (c_N - c_B A_B^{-1} A_N) x_N + c_B A_B^{-1} b \\ \text{s.t.} \quad & x_B + A_B^{-1} A_N x_N = A_B^{-1} b \\ & x \geq 0 \end{aligned} \tag{2.8}$$

Now clearly we see that if $A_B^{-1} b \geq 0$, then setting $x = (x_B, x_N) = (A_B^{-1} b, 0)$ is a feasible solution with objective value $c_B A_B^{-1} b$.

We say that the quantity

$$\tilde{c}_N = c_N - c_B A_B^{-1} A_N.$$

are the *reduced costs*.

Notice that we can re-write the equation above as

$$\begin{aligned} \max \quad & \tilde{c}_N x_N + c_B A_B^{-1} b \\ \text{s.t.} \quad & x_B + A_B^{-1} A_N x_N = A_B^{-1} b \\ & x \geq 0 \end{aligned} \tag{2.9}$$

Hence, viewing this LP from the basis B illuminates that change in objective function as we increase the non-basic variables from 0.

2.3.2. Tableau Based Pivoting

In this section, we discuss how to solve a linear program using a *tableau*. A tableau is just a table to record calculations in a convenient way.

Example 2.3

Solve this linear program using a tabluea based approach.

$$\begin{aligned} \text{Minimize } Z = 2x_1 + 3x_2 & \\ \text{s.t.} \qquad 2x_1 + x_2 & \leq 16 \\ x_1 + 3x_2 & \geq 20 \\ x_1 + x_2 & = 10 \\ x_1, x_2 & \geq 0 \end{aligned} \tag{2.10}$$

We will use the *Big-M* method to solve this problem. We begin by converting the problem into standard form.

$$\begin{aligned}
\text{Maximize } -Z = -2x_1 - 3x_2 - M\bar{x}_5 - M\bar{x}_6 & \\
\text{s.t.} \qquad 2x_1 + x_2 + x_3 &= 16 \\
x_1 + 3x_2 - x_4 + \bar{x}_5 &= 20 \\
x_1 + x_2 + \bar{x}_6 &= 10 \\
x_1, x_2, x_3, x_4, \bar{x}_5, \bar{x}_6 &\geq 0
\end{aligned}$$

Initial Set-up

Basic Variable	Z	Coefficient of: x_1	x_2	x_3	x_4	$\bar{x}_5$	$\bar{x}_6$	Right Side
Z	-1	2	3	0	0	M	M	0
x_3	0	2	1	1	0	0	0	16
$\bar{x}_5$	0	1	3	0	-1	1	0	20
$\bar{x}_6$	0	1	1	0	0	0	1	10

Standard Form

Basic Variable	Z	Coefficient of: x_1	x_2	x_3	x_4	$\bar{x}_5$	$\bar{x}_6$	Right Side
Z	-1	2 - 2 M	3 - 4 M	0	M	0	0	-30 M
x_3	0	2	1	1	0	0	0	16
$\bar{x}_5$	0	1	3	0	-1	1	0	20
$\bar{x}_6$	0	1	1	0	0	0	1	10

Iteration 1 - Let x_2 enter and $\bar{x}_5$ leaves.

Basic Variable	Z	Coefficient of: x_1	x_2	x_3	x_4	$\bar{x}_5$	$\bar{x}_6$	Right Side
Z	-1	1 - 2M/3	0	0	1 - M/3	-1 + 4M/3	0	-20 - 10M/3
x_3	0	2/3	0	1	1/3	-1/3	0	-28/3
x_2	0	1/3	1	0	-1/3	1/3	0	20/3
$\bar{x}_6$	0	2/3	0	0	1/3	-1/3	1	10/3

Iteration 2 - Let x_1 enter and $\bar{x}_6$ leaves.

Basic	Coefficient of:							Right
Variable	Z	x_1	x_2	x_3	x_4	$\bar{x}_5$	$\bar{x}_6$	Side
Z	-1	0	0	0	1/2	-1/2 + M	-3/2 + M	-25
x_3	0	0	0	1	-1/2	1/2	-5/2	1
x_2	0	0	1	0	-1/2	1/2	-1/2	5
x_1	0	1	0	0	1/2	-1/2	3/2	5

We have reached an optimal solution since all coefficients in the objective function are positive. Thus our solution to the initial minimization problem is

$$x_1 = 5, \quad x_2 = 5, \quad Z(5,5) = 25 \qquad \text{with slack } x_3 = 1$$

3. Linear Programming Book - Cheung

Preface

This book covers the fundamentals of linear programming through studying systems of linear inequalities using only basic facts from linear algebra. It is suitable for a crash course on linear programming that emphasizes theoretical aspects of the subject. Discussion on practical solution methods such as the simplex method and interior point methods, though not present in this book, is planned for a future book.

Two excellent references for further study are [**Bertsimas:1997**] and [**Schrijver:1986**].

Notation

The set of real numbers is denoted by $\mathbb{R}$. The set of rational numbers is denoted by $\mathbb{Q}$. The set of integers is denoted by $\mathbb{Z}$.

The set of n-tuples with real entries is denoted by $\mathbb{R}^n$. Similar definitions hold for $\mathbb{Q}^n$ and $\mathbb{Z}^n$.

The set of $m \times n$ matrices (that is, matrices with m rows and n columns) with real entries is denoted $\mathbb{R}^{m\times n}$. Similar definitions hold for $\mathbb{Q}^{m\times n}$ and $\mathbb{Z}^n$.

All n-tuples are written as columns (that is, as $n \times 1$ matrices). An n-tuple is normally represented by a lowercase Roman letter in boldface; for example, $\mathbf{x}$. For an n-tuple $\mathbf{x}$, x_i denotes the ith entry (or component) of $\mathbf{x}$ for $i = 1,\ldots,n$.

Matrices are normally represented by an uppercase Roman letter in boldface; for example, $\mathbf{A}$. The jth column of a matrix $\mathbf{A}$ is denoted by A_j and the (i, j)-entry (that is, the entry in row i and column j) is denoted by a_{ij}.

Scalars are usually represented by lowercase Greek letters; for example, λ, α, β etc.

An n-tuple consisting of all zeros is denoted by 0. The dimension of the tuple is inferred from the context.

For a matrix $\mathbf{A}$, $\mathbf{A}^\mathsf{T}$ denotes the transpose of $\mathbf{A}$. For an n-tuple $\mathbf{x}$, $\mathbf{x}^\mathsf{T}$ denotes the transpose of $\mathbf{x}$.

If $\mathbf{A}$ and $\mathbf{B}$ are $m \times n$ matrices, $\mathbf{A} \geq \mathbf{B}$ means $a_{ij} \geq b_{ij}$ for all $i = 1, \ldots, m,\ j = 1, \ldots, n$. Similar definitions hold for $\mathbf{A} \leq \mathbf{B}$, $\mathbf{A} = \mathbf{B}$, $\mathbf{A} < \mathbf{B}$ and $\mathbf{A} > \mathbf{B}$. In particular, if $\mathbf{u}$ and $\mathbf{v}$ are n-tuples, $\mathbf{u} \geq \mathbf{v}$ means $u_i \geq v_i$ for $i = 1, \ldots, n$ and $\mathbf{u} > 0$ means $u_i > 0$ for $i = 1, \ldots, n$.

Superscripts in brackets are used for indexing tuples. For example, we can write $\mathbf{u}^{(1)}, \mathbf{u}^{(2)} \in \mathbb{R}^3$. Then $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$ are elements of $\mathbb{R}^3$. The second entry of $\mathbf{u}^{(1)}$ is denoted by $u_2^{(1)}$.

3.1 Graphical example

To motivate the subject of linear programming (LP), we begin with a planning problem that can be solved graphically.

Example 3.1: Lemonade Vendor

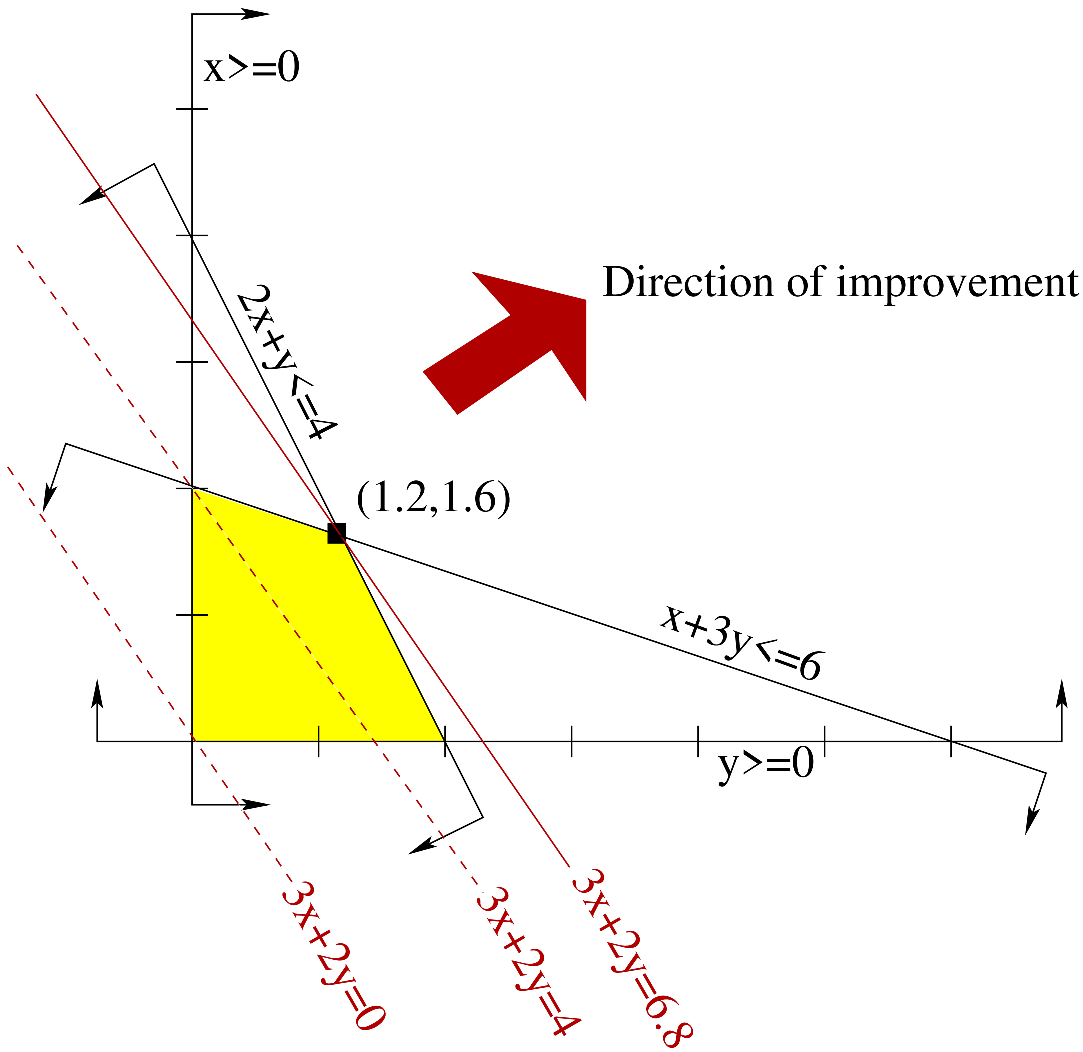
Say you are a vendor of lemonade and lemon juice. Each unit of lemonade requires 1 lemon and 2 litres of water. Each unit of lemon juice requires 3 lemons and 1 litre of water. Each unit of lemonade gives a profit of three dollars. Each unit of lemon juice gives a profit of two dollars. You have 6 lemons and 4 litres of water available. How many units of lemonade and lemon juice should you make to maximize profit?

If we let x denote the number of units of lemonade to be made and let y denote the number of units of lemon juice to be made, then the profit is given by $3x + 2y$ dollars. We call $3x + 2y$ the objective function. Note that there are a number of constraints that x and y must satisfied. First of all, x and y should be nonnegative. The number of lemons needed to make x units of lemonade and y units of lemon juice is $x + 3y$ and cannot exceed 6. The number of litres of water needed to make x units of lemonade and y units of lemon juice is $2x + y$ and cannot exceed 4. Hence, to determine the maximum profit, we need to maximize $3x + 2y$ subject to x and y satisfying the constraints $x + 3y \leq 6$, $2x + y \leq 4$, $x \geq 0$, and $y \geq 0$.

A more compact way to write the problem is as follows:

$$
\begin{array}{rrcrcr}
\textit{maximize} & 3x & + & 2y & & \\
\textit{subject to} & x & + & 3y & \leq & 6 \\
 & 2x & + & y & \leq & 4 \\
 & x & & & \geq & 0 \\
 & & & y & \geq & 0.
\end{array}
$$

We can solve this maximization problem graphically as follows. We first sketch the set of $\begin{bmatrix} x \\ y \end{bmatrix}$ satisfying the constraints, called the feasible region, on the (x, y)-plane. We then take the objective function $3x + 2y$ and turn it into an equation of a line $3x + 2y = z$ where z is a parameter. Note that as the value of z increases, the line defined by the equation $3x + 2y = z$ moves in the direction of the normal vector $\begin{bmatrix} 3 \\ 2 \end{bmatrix}$. We call this direction the direction of improvement. Determining the maximum value of the objective function, called the optimal value, subject to the contraints amounts to finding the maximum value of z so that the line defined by the equation $3x+2y=z$ still intersects the feasible region.



In the figure above, the lines with z at 0, 4 and 6.8 have been drawn. From the picture, we can see that if z is greater than 6.8, the line defined by $3x+2y=z$ will not intersect the feasible region. Hence, the profit cannot exceed 6.8 dollars.

As the line $3x+2y=6.8$ does intersect the feasible region, 6.8 is the maximum value for the objective function. Note that there is only one point in the feasible region that intersects the line $3x+2y=6.8$, namely $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1.2 \\ 1.6 \end{bmatrix}$. In other words, to maximize profit, we want to make 1.2 units of lemonade and 1.6 units of lemon juice.

The above solution method can hardly be regarded as rigorous because we relied on a picture to conclude that $3x+2y \leq 6.8$ for all $\begin{bmatrix} x \\ y \end{bmatrix}$ satisfying the constraints. But we can actually show this *algebraically*.

Note that multiplying both sides of the constraint $x+3y \leq 6$ gives $0.2x+0.6y \leq 1.2$, and multiplying both sides of the constraint $2x+y \leq 4$ gives $2.8x+1.4y \leq 5.6$. Hence, any $\begin{bmatrix} x \\ y \end{bmatrix}$ that satisfies both $x+3y \leq 6$ and $2x+y \leq 4$ must also satisfy $(0.2x+0.6y)+(2.8x+1.4y) \leq 1.2+5.6$, which simplifies to $3x+2y \leq 6.8$ as desired! (Here, we used the fact that if $a \leq b$ and $c \leq d$, then $a+c \leq b+d$.)

Now, one might ask if it is always possible to find an algebraic proof like the one above for similar problems. If the answer is yes, how does one find such a proof? We will see answers to this question later on.

Before we end this segment, let us consider the following problem:

$$\begin{array}{rrcrcl} \text{minimize} & -2x & + & y & & \\ \text{subject to} & -x & + & y & \leq & 3 \\ & x & - & 2y & \leq & 2 \\ & x & & & \geq & 0 \\ & & & y & \geq & 0. \end{array}$$

Note that for any $t \geq 0$, $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} t \\ t \end{bmatrix}$ satisfies all the constraints. The value of the objective function at $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} t \\ t \end{bmatrix}$ is $-t$. As $t \to \infty$, the value of the objective function tends to $-\infty$. Therefore, there is no minimum value for the objective function. The problem is said to be unbounded. Later on, we will see how to detect unboundedness algorithmically.

As an exercise, check that unboundedness can also be established by using $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2t+2 \\ t \end{bmatrix}$ for $t \geq 0$.

Exercises

1. Sketch all $\begin{bmatrix} x \\ y \end{bmatrix}$ satisfying

$$x - 2y \leq 2$$

   on the (x,y)-plane.

2. Determine the optimal value of

$$\begin{array}{rl} \text{Minimize} & x+y \\ \text{Subject to} & 2x+y \geq 4 \\ & x+3y \geq 1. \end{array}$$

3. Show that the problem

$$\begin{array}{rl} \text{Minimize} & -x+y \\ \text{Subject to} & 2x-y \geq 0 \\ & x+3y \geq 3 \end{array}$$

   is unbounded.

4. Suppose that you are shopping for dietary supplements to satisfy your required daily intake of 0.40mg of nutrient M and 0.30mg of nutrient N. There are three popular products on the market. The costs and the amounts of the two nutrients are given in the following table:

	Product 1	Product 2	Product 3
Cost	\$27	\$31	\$24
Daily amount of M	0.16 mg	0.21 mg	0.11 mg
Daily amount of N	0.19 mg	0.13 mg	0.15 mg

You want to determine how much of each product you should buy so that the daily intake requirements of the two nutrients are satisfied at minimum cost. Formulate your problem as a linear programming problem, assuming that you can buy a fractional number of each product.

Solutions

1. The points (x,y) satisfying $x-2y \leq 2$ are precisely those above the line passing through $(2,0)$ and $(0,-1)$.

2. We want to determine the minimum value z so that $x+y=z$ defines a line that has a nonempty intersection with the feasible region. However, we can avoid referring to a sketch by setting $x=z-y$ and substituting for x in the inequalities to obtain:

$$\begin{aligned} 2(z-y)+y &\geq 4 \\ (z-y)+3y &\geq 1, \end{aligned}$$

or equivalently,

$$\begin{aligned} z &\geq 2+\frac{1}{2}y \\ z &\geq 1-2y, \end{aligned}$$

Thus, the minimum value for z is $\min\{2+\frac{1}{2}y, 1-2y\}$, which occurs at $y=-\frac{2}{5}$. Hence, the optimal value is $\frac{9}{5}$.

We can verify our work by doing the following. If our calculations above are correct, then an optimal solution is given by $x=\frac{11}{5}$, $y=-\frac{2}{5}$ since $x=z-y$. It is easy to check that this satisfies both inequalities and therefore is a feasible solution.

Now, taking $\frac{2}{5}$ times the first inequality and $\frac{1}{5}$ times the second inequality, we can infer the inequality $x+y \geq \frac{9}{5}$. The left-hand side of this inequality is precisely the objective function. Hence, no feasible solution can have objective function value less than $\frac{9}{5}$. But $x=\frac{11}{5}$, $y=-\frac{2}{5}$ is a feasible solution with objective function value equal to $\frac{9}{5}$. As a result, it must be an optimal solution.

Remark. We have not yet discussed how to obtain the multipliers $\frac{2}{5}$ and $\frac{1}{5}$ for inferring the inequality $x+y \geq \frac{9}{5}$. This is an issue that will be taken up later. In the meantime, think about how one could have obtained these multipliers for this particular exercise.

3. We could glean some insight by first making a sketch on the (x,y)-plane.

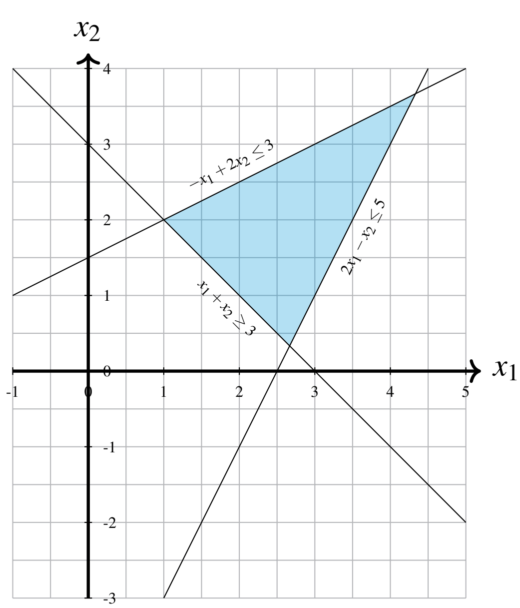
   The line defined by $-x+y=z$ has x-intercept $-z$. Note that for $z \leq -3$, $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -z \\ 0 \end{bmatrix}$ satisfies both inequalities and the value of the objective function at $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -z \\ 0 \end{bmatrix}$ is z. Hence, there is no lower bound on the value of objective function.

4. Let x_i denote the amount of Product i to buy for $i = 1,2,3$. Then, the problem can be formulated as

$$\begin{array}{rrcrcrcl} \text{minimize} & 27x_1 & + & 31x_2 & + & 24x_3 & & \\ \text{subject to} & 0.16x_1 & + & 0.21x_2 & + & 0.11x_3 & \geq & 0.30 \\ & 0.19x_1 & + & 0.13x_2 & + & 0.15x_3 & \geq & 0.40 \\ & x_1 & , & x_2 & , & x_3 & \geq & 0. \end{array}$$

Remark. If one cannot buy fractional amounts of the products, the problem can be formulated as

$$\begin{array}{rrcrcrcl} \text{minimize} & 27x_1 & + & 31x_2 & + & 24x_3 & & \\ \text{subject to} & 0.16x_1 & + & 0.21x_2 & + & 0.11x_3 & \geq & 0.30 \\ & 0.19x_1 & + & 0.13x_2 & + & 0.15x_3 & \geq & 0.40 \\ & x_1 & , & x_2 & , & x_3 & \geq & 0. \\ & x_1 & , & x_2 & , & x_3 & \in & \mathbb{Z}. \end{array}$$



[1]

[1]https://tex.stackexchange.com/questions/75933/how-to-draw-the-region-of-inequality

3.2 Definitions

The following is an example of a problem in **linear programming**:

$$\begin{array}{rl} \text{Maximize} & x+y-2z \\ \text{Subject to} & 2x+y+z \leq 4 \\ & 3x-y+z=0 \\ & x,y,z \geq 0 \end{array}$$

Solving this problem means finding real values for the **variables** x,y,z satisfying the **constraints** $2x+y+z \leq 4$, $3x-y+z=0$, and $x,y,z \geq 0$ that gives the maximum possible value (if it exists) for the **objective function** $x+y-2z$.

For example, $\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ satisfies all the constraints and is called a **feasible solution**. Its **objective function value**, obtained by evaluating the objective function at $\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$, is $0+1-2(1)=-1$. The set of feasible solutions to a linear programming problem is called the **feasible region**.

More formally, a linear programming problem is an optimization problem of the following form:

$$\begin{array}{rl} \text{Maximize (or Minimize)} & \displaystyle\sum_{j=1}^{n} c_j x_j \\ \text{Subject to} & P_i(x_1,\ldots,x_n) \quad i=1,\ldots,m \end{array}$$

where m and n are positive integers, $c_j \in \mathbb{R}$ for $j=1,\ldots,n$, and for each $i=1,\ldots,m$, $P_i(x_1,\ldots,x_n)$ is a **linear constraint** on the **(decision) variables** $x_1,\ldots,x_n$ having one of the following forms:

- $a_1x_1+\cdots+a_nx_n \geq \beta$
- $a_1x_1+\cdots+a_nx_n \leq \beta$
- $a_1x_1+\cdots a_nx_n = \beta$

where $\beta, a_1,\ldots,a_n \in \mathbb{R}$. To save writing, the word "Minimize" ("Maximize") is replaced with "min" ("max") and "Subject to" is abbreviated as "s.t.".

A feasible solution $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ that gives the maximum possible objective function value in the case of a maximization problem is called an **optimal solution** and its objective function value is the **optimal value** of the problem.

The following example shows that it is possible to have multiple optimal solutions:

$$\begin{array}{rl} \max & x+y \\ \text{s.t.} & 2x+2y \leq 1 \end{array}$$

The constraint says that $x+y$ cannot exceed $\frac{1}{2}$. Now, both $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix}$ and $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{2} \end{bmatrix}$ are feasible solutions having objective function value $\frac{1}{2}$. Hence, they are both optimal solutions. (In fact, this problem has infinitely many optimal solutions. Can you specify all of them?)

Not all linear programming problems have optimal solutions. For example, a problem can have no feasible solution. Such a problem is said to be **infeasible**. Here is an example of an infeasible problem:

$$\begin{array}{rl} \min & x \\ \text{s.t.} & x \leq 1 \\ & x \geq 2 \end{array}$$

There is no value for x that is at the same time at most 1 and at least 2.

Even if a problem is not infeasible, it might not have an optimal solution as the following example shows:

$$\begin{array}{rl} \min & x \\ \text{s.t.} & x \leq 0 \end{array}$$

Note that now matter what real number M we are given, we can always find a feasible solution whose objective function value is less than M. Such a problem is said to be **unbounded**. (For a maximization problem, it is unbounded if one can find feasible solutions who objective function value is larger than any given real number.)

So far, we have seen that a linear programming problem can have an optimal solution, be infeasible, or be unbounded. Is it possible for a linear programming problem to be not infeasible, not unbounded, and with no optimal solution?
The following optimization problem, though not a linear programming problem, is not infeasible, not unbounded, and has no optimal solution:

$$\begin{array}{rl} \min & 2^x \\ \text{s.t.} & x \leq 0 \end{array}$$

The objective function value is never negative and can get arbitrarily close to 0 but can never attain 0.

A main result in linear programming states that if a linear programming problem is not infeasible and is not unbounded, then it must have an optimal solution. This result is known as the **Fundamental Theorem of Linear Programming** (Theorem **??**) and we will see a proof of this importan result. In the meantime, we will consider the seemingly easier problem of determining if a system of linear constraints has a solution.

Exercises

1. Determine all values of a such that the problem

$$\begin{array}{rl} \min & x+y \\ \text{s.t.} & -3x+y \geq a \\ & 2x-y \geq 0 \\ & x+2y \geq 2 \end{array}$$

   is infeasible.

2. Show that the problem

$$\begin{array}{rl} \min & 2^x \cdot 4^y \\ \text{s.t.} & e^{-3x+y} \geq 1 \\ & |2x-y| \leq 4 \end{array}$$

   can be solved by solving a linear programming problem.

Solutions

1. Adding the first two inequalities gives $-x \geq a$. Adding 2 times the second inequality and the third inequality gives $5x \geq 2$, implying that $x \geq \frac{2}{5}$. Hence, if $a > -\frac{2}{5}$, there is no solution.

   Note that if $a \leq -\frac{2}{5}$, then $(x,y) = \left(\frac{2}{5}, \frac{4}{5}\right)$ satisfies all the inequalities. Hence, the problem is infeasible if and only if $a > -\frac{2}{5}$.

2. Note that the constraint $|2x-y| \leq 4$ is equivalent to the constraints $2x-y \leq 4$ and $2x-y \geq -4$ taken together, and the constraint $e^{-3x+y} \geq 1$ is equivalent to $-3x+y \geq 0$. Hence, we can rewrite the problem with linear constraints.

   Finally, minimizing $2^x \cdot 4^y$ is the same as minimizing 2^{x+2y}, which is equivalent to minimizing $x+2y$.

3.3 Farkas' Lemma

A well-known result in linear algebra states that a system of linear equations $\mathbf{A}\mathbf{x} = \mathbf{b}$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ is a tuple of variables, has no solution if and only if there exists $\mathbf{y} \in \mathbb{R}^m$ such that $\mathbf{y}^\mathsf{T}\mathbf{A} = 0$ and $\mathbf{y}^\mathsf{T}\mathbf{b} \neq 0$.

It is easily seen that if such a $\mathbf{y}$ exists, then the system $\mathbf{A}\mathbf{x} = \mathbf{b}$ cannot have a solution. (Simply multiply both sides of $\mathbf{A}\mathbf{x} = \mathbf{b}$ on the left by $\mathbf{y}^\mathsf{T}$.) However, proving the converse requires a bit of work. A standard elementary proof involves using Gauss-Jordan elimination to reduce the original system to an equivalent system $\mathbf{Qx} = \mathbf{d}$ such that $\mathbf{Q}$ has a row of zero, say in row i, with $\mathbf{d}_i \neq 0$. The process can be captured by a square matrix $\mathbf{M}$ satisfying $\mathbf{MA} = \mathbf{Q}$. We can then take $\mathbf{y}^\mathsf{T}$ to be the ith row of $\mathbf{M}$.

An analogous result holds for systems of linear inequalities. The following result is one of the many variants of a result known as the **Farkas' Lemma**:

Theorem 3.2: Farkas' Lemma

With $\mathbf{A}$, $\mathbf{x}$, and $\mathbf{b}$ as above, the system $\mathbf{Ax} \geq \mathbf{b}$ has no solution if and only if there exists $\mathbf{y} \in \mathbb{R}^m$ such that

$$\mathbf{y} \geq 0,\ \mathbf{y}^\mathsf{T}\mathbf{A} = 0,\ \mathbf{y}^\mathsf{T}\mathbf{b} > 0.$$

In other words, the system $\mathbf{Ax} \geq \mathbf{b}$ has no solution if and only if one can infer the inequality $0 \geq \gamma$ for some $\gamma > 0$ by taking a nonnegative linear combination of the inequalities.

This result essentially says that there is always a certificate (the m-tuple $\mathbf{y}$ with the prescribed properties) for the infeasibility of the system $\mathbf{Ax} \geq \mathbf{b}$. This allows third parties to verify the claim of infeasibility without having to solve the system from scratch.

Example 3.3

For the system

$$\begin{aligned} 2x - y + z &\geq 2 \\ -x + y - z &\geq 0 \\ -y + z &\geq 0, \end{aligned}$$

adding two times the second inequality and the third inequality to the first inequality gives $0 \geq 2$. Hence, $\mathbf{y} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$ is a certificate of infeasibility for this example.

We now give a proof of Theorem **??**. It is easy to see that if such a $\mathbf{y}$ exists, then the system $\mathbf{Ax} \geq \mathbf{b}$ has no solution.

3.4 Fundamental Theorem of Linear Programming

Having used Fourier-Motzkin elimination to solve a linear programming problem, we now will go one step further and use the same technique to prove the following important result.

Theorem 3.4: Fundamental Theorem of Linear Programming

For any given linear programming problem, exactly one of the following holds:

1. *the problem is infeasible;*
2. *the problem is unbounded;*
3. *the problem has an optimal solution.*

Proof. Without loss of generality, we may assume that the linear programming problem is of the form

$$\begin{array}{rl} \min & \mathbf{c}^\mathsf{T}\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \geq \mathbf{b} \end{array} \tag{3.1}$$

where m and n are positive integers, $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$, and $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ is a tuple of variables. Indeed, any linear programming problem can be converted to a linear programming problem in the form of (**??**) having the same feasible region and optimal solution set. To see this, note that a constraint of the form $\mathbf{a}^\mathsf{T}\mathbf{x} \leq \beta$ can be written as $-\mathbf{a}^\mathsf{T}\mathbf{x} \geq -\beta$; a constraint of the form $\mathbf{a}^\mathsf{T}\mathbf{x} = \beta$ written as a pair of constraints $\mathbf{a}^\mathsf{T}\mathbf{x} \geq \beta$ and $-\mathbf{a}^\mathsf{T}\mathbf{x} \geq -\beta$; and a maximization problem is equivalent to the problem that minimizes the negative of the objective function subject to the same constraints.

Suppose that (**??**) is not infeasible. Form the system

$$\begin{aligned} z - \mathbf{c}^\mathsf{T}\mathbf{x} &\geq 0 \\ -z + \mathbf{c}^\mathsf{T}\mathbf{x} &\geq 0 \\ \mathbf{A}\mathbf{x} &\geq \mathbf{b}. \end{aligned} \tag{3.2}$$

Solving (**??**) is equivalent to finding among all the solutions to (**??**) one that minimizes z, if it exists. Eliminating the variables $x_1, \ldots, x_n$ (in any order) using Fourier-Motzkin elimination gives a system of linear inequalities (S) containing at most the variable z. By scaling, we may assume that the each coefficient of z in (S) is 1, -1, or 0. Note that any z satisfying (S) can be extended to a solution to (**??**) and the z value from any solution to (**??**) must satisfy (S).

That (**??**) is not unbounded implies that (S) must contain an inequality of the form $z \geq \beta$ for some $\beta \in \mathbb{R}$. (Why?) Let all the inequalites in which the coefficient of z is positive be

$$z \geq \beta_i$$

where $\beta_i \in \mathbb{R}$ for $i = 1, \ldots, p$ for some positive integer p. Let $\gamma = \max\{\beta_1, \ldots, \beta_p\}$. Then for any solution x, z to (**??**), z is at least γ. But we can set $z = \gamma$ and extend it to a solution to (**??**). Hence, we obtain an optimal solution for (**??**) and γ is the optimal value. This completes the proof of the theorem.

□

Remark. We can construct multipliers to infer the inequality $\mathbf{c}^\mathsf{T}\mathbf{x} \geq \gamma$ from the system $\mathbf{A}\mathbf{x} \geq \mathbf{b}$. Because we obtained the inequality $z \geq \gamma$ using Fourier-Motzkin elimination, there must exist real numbers $\alpha, \beta, y_1^*, \ldots, y_m^* \geq 0$ such that

$$\begin{bmatrix} \alpha & \beta & y_1^* & \cdots & y_m^* \end{bmatrix} \begin{bmatrix} 1 & -\mathbf{c}^\mathsf{T} \\ -1 & \mathbf{c}^\mathsf{T} \\ 0 & \mathbf{A} \end{bmatrix} \begin{bmatrix} z \\ \mathbf{x} \end{bmatrix} \geq \begin{bmatrix} \alpha & \beta & y_1^* & \cdots & y_m^* \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \mathbf{b} \end{bmatrix}$$

is identically $z \geq \gamma$. Note that we must have $\alpha - \beta = 1$ and

$$\mathbf{y}^* \geq 0,\ {\mathbf{y}^*}^\mathsf{T}\mathbf{A} = \mathbf{c}^\mathsf{T},\ \text{and } {\mathbf{y}^*}^\mathsf{T}\mathbf{b} = \gamma$$

where $\mathbf{y}^* = [y_1^*, \ldots, y_m^*]^\mathsf{T}$. Hence, $y_1^*, \ldots, y_m^*$ are the desired multipliers.,

The significance of the fact that we can infer $\mathbf{c}^\mathsf{T}\mathbf{x} \geq \gamma$ where γ will be discussed in more details when we look at duality theory for linear programming.

Exercises

1. Determine the optimal value of the following linear programming problem:

$$\begin{array}{rl} \min & x \\ \text{s.t.} & x + y \geq 2 \\ & x - 2y + z \geq 0 \\ & y - 2z \geq -1. \end{array}$$

2. Determine if the following linear programming problem has an optimal solution:

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

3. A set $S \subset \mathbb{R}^n$ is said to be bounded if there exists a real number $M > 0$ such that for every $\mathbf{x} \in S$, $|x_i| < M$ for all $i = 1, \ldots, n$. Prove that every linear programming problem with a bounded nonempty feasible region has an optimal solution.

Solutions

1. The problem is equivalent to determining the minimum value for x among all x, y, z satisfying

$$\begin{array}{rl} x + y \geq 2 & (1) \\ x - 2y + z \geq 0 & (2) \\ y - 2z \geq -1. & (3) \end{array}$$

We use Fourier-Motzkin Elimination Method to eliminate z. Multiplying (3) by $\frac{1}{2}$, we get

$$\begin{aligned} x+y &\geq 2 & (1)\\ x-2y+z &\geq 0 & (2)\\ \tfrac{1}{2}y-z &\geq -\tfrac{1}{2}. & (4) \end{aligned}$$

Eliminating z, we obtain

$$\begin{aligned} x+y &\geq 2 & (1)\\ x-\tfrac{3}{2}y &\geq -\tfrac{1}{2} & (5) \end{aligned}$$

where (5) is given by $(2)+(4)$.

Multiplying (5) by $\frac{2}{3}$, we get

$$\begin{aligned} x+y &\geq 2 & (1)\\ \tfrac{2}{3}x-y &\geq -\tfrac{1}{3} & (6) \end{aligned}$$

Eliminating y, we get

$$\tfrac{5}{3}x \geq \tfrac{5}{3} \qquad (7)$$

where (7) is given by $(1)+(6)$. Multiplying (7) by $\frac{3}{5}$, we obtain $x \geq 1$. Hence, the minimum possible value for x is 1.

Note that setting $x=1$, the system (1) and (6) forces $y=1$. And (2) and (3) together force $z=1$. One can check that $(x,y,z)=(1,1,1)$ is a feasible solution.

Remark. Note that the inequality $x \geq 1$ is given by

$$\begin{aligned} \frac{3}{5}(7) &\Leftarrow \frac{3}{5}(1)+\frac{3}{5}(6)\\ &\Leftarrow \frac{3}{5}(1)+\frac{2}{5}(5)\\ &\Leftarrow \frac{3}{5}(1)+\frac{2}{5}(2)+\frac{2}{5}(4)\\ &\Leftarrow \frac{3}{5}(1)+\frac{2}{5}(2)+\frac{1}{5}(3) \end{aligned}$$

2. It suffices to determine if there exists a minimum value for z among all the solutions to the system

$$\begin{aligned} z-x_1-2x_2 &\geq 0 & (1)\\ -z+x_1+2x_2 &\geq 0 & (2)\\ x_1+3x_2 &\geq 4 & (3)\\ -x_1+x_2 &\geq 0 & (4) \end{aligned}$$

Using Fourier-Motzkin elimination to eliminate x_1, we obtain:

$$\begin{aligned} (1)+(2):&\quad & 0 &\geq 0 &\\ (1)+(3):&\quad & z+x_2 &\geq 4 & (5)\\ (2)+(4):&\quad & -z+3x_2 &\geq 0 & (6)\\ (3)+(4):&\quad & 4x_2 &\geq 4 & (7) \end{aligned}$$

Note that all the coefficients of x_2 is nonnegative. Hence, eliminating x_2 will result in a system with no constraints. Therefore, there is no lower bound on the value of z. In particular, if $z = t$ for $t \leq 0$, then from (5)–(6), we need $x_2 \geq 4 - t$, $3x_2 \geq t$, and $x_2 \geq 1$. Hence, we can set $x_2 = 4 - t$ and $x_1 = -8 + 3t$. This gives a feasible solution for all $t \leq 0$ with objective function value that approaches $-\infty$ as $t \to -\infty$. Hence, the linear programming problem is unbounded.

3. Let (P) denote a linear programming problem with a bounded nonempty feasible region with objective function $\mathbf{c}^\mathsf{T}\mathbf{x}$. By assumption, (P) is not infeasible. Note that (P) is not unbounded because $|\mathbf{c}^\mathsf{T}\mathbf{x}| \leq \sum_i |c_i||x_i| \leq M \sum_i |c_i|$. Thus, by Theorem **??**, (P) has an optimal solution.

3.5 Linear programming duality

Consider the following problem:

$$\begin{array}{rl} \min & \mathbf{c}^\mathsf{T}\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \geq \mathbf{b}. \end{array} \tag{3.1}$$

In the remark at the end of Chapter **??**, we saw that if (**??**) has an optimal solution, then there exists $\mathbf{y}^* \in \mathbb{R}^m$ such that $\mathbf{y}^* \geq 0$, ${\mathbf{y}^*}^\mathsf{T}\mathbf{A} = \mathbf{c}^\mathsf{T}$, and ${\mathbf{y}^*}^\mathsf{T}\mathbf{b} = \gamma$ where γ denotes the optimal value of (**??**).

Take any $\mathbf{y} \in \mathbb{R}^m$ satisfying $\mathbf{y} \geq 0$ and $\mathbf{y}^\mathsf{T}\mathbf{A} = \mathbf{c}^\mathsf{T}$. Then we can infer from $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ the inequality $\mathbf{y}^\mathsf{T}\mathbf{A}\mathbf{x} \geq \mathbf{y}^\mathsf{T}\mathbf{b}$, or more simply, $\mathbf{c}^\mathsf{T}\mathbf{x} \geq \mathbf{y}^\mathsf{T}\mathbf{b}$. Thus, for any such $\mathbf{y}$, $\mathbf{y}^\mathsf{T}\mathbf{b}$ gives a lower bound for the objective function value of any feasible solution to (**??**). Since γ is the optimal value of (P), we must have $\gamma \geq \mathbf{y}^\mathsf{T}\mathbf{b}$.

As ${\mathbf{y}^*}^\mathsf{T}\mathbf{b} = \gamma$, we see that γ is the optimal value of

$$\begin{array}{rl} \max & \mathbf{y}^\mathsf{T}\mathbf{b} \\ \text{s.t.} & \mathbf{y}^\mathsf{T}\mathbf{A} = \mathbf{c}^\mathsf{T} \\ & \mathbf{y} \geq 0. \end{array} \tag{3.2}$$

Note that (**??**) is a linear programming problem! We call it the **dual problem** of the **primal problem** (**??**). We say that the dual variable y_i is **associated** with the constraint ${\mathbf{a}^{(i)}}^\mathsf{T}\mathbf{x} \geq b_i$ where ${\mathbf{a}^{(i)}}^\mathsf{T}$ denotes the ith row of $\mathbf{A}$.

In other words, we define the dual problem of (**??**) to be the linear programming problem (**??**). In the discussion above, we saw that if the primal problem has an optimal solution, then so does the dual problem and the optimal values of the two problems are equal. Thus, we have the following result:

Theorem 3.5: strong-duality-special

Suppose that (**??**) *has an optimal solution. Then* (**??**) *also has an optimal solution and the optimal values of the two problems are equal.*

At first glance, requiring all the constraints to be $\geq$-inequalities as in (**??**) before forming the dual problem seems a bit restrictive. We now see how the dual problem of a primal problem in general form can be defined. We first make two observations that motivate the definition.

Observation 1

Suppose that our primal problem contains a mixture of all types of linear constraints:

$$\begin{array}{rl} \min & \mathbf{c}^\mathsf{T}\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & \mathbf{A}'\mathbf{x} \leq \mathbf{b}' \\ & \mathbf{A}''\mathbf{x} = \mathbf{b}'' \end{array} \tag{3.3}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{A}' \in \mathbb{R}^{m'\times n}$, $\mathbf{b}' \in \mathbb{R}^{m'}$, $\mathbf{A}'' \in \mathbb{R}^{m''\times n}$, and $\mathbf{b}'' \in \mathbb{R}^{m''}$.

We can of course convert this into an equivalent problem in the form of (**??**) and form its dual.
However, if we take the point of view that the function of the dual is to infer from the constraints of (**??**) an inequality of the form $\mathbf{c}^\mathsf{T}\mathbf{x} \geq \gamma$ with γ as large as possible by taking an appropriate linear combination of the constraints, we are effectively looking for $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{y} \geq 0$, $\mathbf{y}' \in \mathbb{R}^{m'}$, $\mathbf{y}' \leq 0$, and $\mathbf{y}'' \in \mathbb{R}^{m''}$, such that

$$\mathbf{y}^\mathsf{T}\mathbf{A} + \mathbf{y}'^\mathsf{T}\mathbf{A}' + \mathbf{y}''^\mathsf{T}\mathbf{A}'' = \mathbf{c}^\mathsf{T}$$

with $\mathbf{y}^\mathsf{T}\mathbf{b} + \mathbf{y}'^\mathsf{T}\mathbf{b}' + \mathbf{y}''^\mathsf{T}\mathbf{b}''$ to be maximized.

(The reason why we need $\mathbf{y}' \leq 0$ is because inferring a $\geq$-inequality from $\mathbf{A}'\mathbf{x} \leq \mathbf{b}'$ requires nonpositive multipliers. There is no restriction on $\mathbf{y}''$ because the constraints $\mathbf{A}''\mathbf{x} = \mathbf{b}''$ are equalities.)

This leads to the dual problem:

$$\begin{array}{rl} \max & \mathbf{y}^\mathsf{T}\mathbf{b} + \mathbf{y}'^\mathsf{T}\mathbf{b}' + \mathbf{y}''^\mathsf{T}\mathbf{b}'' \\ \text{s.t.} & \mathbf{y}^\mathsf{T}\mathbf{A} + \mathbf{y}'^\mathsf{T}\mathbf{A}' + \mathbf{y}''^\mathsf{T}\mathbf{A}'' = \mathbf{c}^\mathsf{T} \\ & \quad \mathbf{y} \geq 0 \\ & \quad \mathbf{y}' \leq 0. \end{array} \tag{3.4}$$

In fact, we could have derived this dual by applying the definition of the dual problem to

$$\begin{array}{rl} \min & \mathbf{c}^\mathsf{T}\mathbf{x} \\ \text{s.t.} & \begin{bmatrix} \mathbf{A} \\ -\mathbf{A}' \\ \mathbf{A}'' \\ -\mathbf{A}'' \end{bmatrix} \mathbf{x} \geq \begin{bmatrix} \mathbf{b} \\ -\mathbf{b}' \\ \mathbf{b}'' \\ -\mathbf{b}'' \end{bmatrix}, \end{array}$$

which is equivalent to (**??**). The details are left as an exercise.

Observation 2

Consider the primal problem of the following form:

$$\begin{array}{rl} \min & \mathbf{c}^\mathsf{T}\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & x_i \geq 0 \;\; i \in P \\ & x_i \leq 0 \;\; i \in N \end{array} \tag{3.5}$$

where P and N are disjoint subsets of $\{1,\ldots,n\}$. In other words, constraints of the form $x_i \geq 0$ or $x_i \leq 0$ are separated out from the rest of the inequalities.

Forming the dual of (**??**) as defined under Observation 1, we obtain the dual problem

$$\begin{array}{rll} \max & \mathbf{y}^\mathsf{T}\mathbf{b} & \\ \text{s.t.} & \mathbf{y}^\mathsf{T}\mathbf{a}^{(i)} = c_i & i \in \{1,\ldots,n\} \backslash (P \cup N) \\ & \mathbf{y}^\mathsf{T}\mathbf{a}^{(i)} + p_i = c_i & i \in P \\ & \mathbf{y}^\mathsf{T}\mathbf{a}^{(i)} + q_i = c_i & i \in N \\ & p_i \geq 0 & i \in P \\ & q_i \leq 0 & i \in N \end{array} \tag{3.6}$$

where $\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$. Note that this problem is equivalent to the following without the variables p_i, $i \in P$ and q_i, $i \in N$:

$$\begin{array}{rll} \max & \mathbf{y}^\mathsf{T}\mathbf{b} & \\ \text{s.t.} & \mathbf{y}^\mathsf{T}\mathbf{a}^{(i)} = c_i & i \in \{1,\ldots,n\} \backslash (P \cup N) \\ & \mathbf{y}^\mathsf{T}\mathbf{a}^{(i)} \leq c_i & i \in P \\ & \mathbf{y}^\mathsf{T}\mathbf{a}^{(i)} \geq c_i & i \in N, \end{array} \tag{3.7}$$

which can be taken as the dual problem of (**??**) instead of (**??**). The advantage here is that it has fewer variables than (**??**).

Hence, the dual problem of

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

is simply

$$\begin{array}{rl} \max & \mathbf{y}^\mathsf{T}\mathbf{b} \\ \text{s.t.} & \mathbf{y}^\mathsf{T}\mathbf{A} \leq \mathbf{c}^\mathsf{T} \\ & \mathbf{y} \geq 0. \end{array}$$

As we can see from bove, there is no need to associate dual variables to constraints of the form $x_i \geq 0$ or $x_i \leq 0$ provided we have the appropriate types of constraints in the dual problem. Combining all the observations lead to the definition of the dual problem for a primal problem in general form as discussed next.

3.5.1. The dual problem

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$. Let ${\mathbf{a}^{(i)}}^\mathsf{T}$ denote the ith row of $\mathbf{A}$. Let $\mathbf{A}_j$ denote the jth column of $\mathbf{A}$.

Let (P) denote the minimization problem with variables in the tuple $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ given as follows:

- The objective function to be minimized is $\mathbf{c}^\mathsf{T}\mathbf{x}$
- The constraints are

$$ {\mathbf{a}^{(i)}}^\mathsf{T}\mathbf{x} \sqcup_i b_i $$

  where $\sqcup_i$ is $\leq$, $\geq$, or $=$ for $i = 1, \ldots, m$.
- For each $j \in \{1, \ldots, n\}$, x_j is constrained to be nonnegative, nonpositive, or free (i.e. not constrained to be nonnegative or nonpositive.)

Then the **dual problem** is defined to be the maximization problem with variables in the tuple $\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$ given as follows:

- The objective function to be maximized is $\mathbf{y}^\mathsf{T}\mathbf{b}$
- For $j = 1, \ldots, n$, the jth constraint is

$$ \begin{cases} \mathbf{y}^\mathsf{T}\mathbf{A}_j \leq c_j & \text{if } x_j \text{ is constrained to be nonnegative} \\ \mathbf{y}^\mathsf{T}\mathbf{A}_j \geq c_j & \text{if } x_j \text{ is constrained to be nonpositive} \\ \mathbf{y}^\mathsf{T}\mathbf{A}_j = c_j & \text{if } x_j \text{ is free.} \end{cases} $$

- For each $i \in \{1, \ldots, m\}$, y_i is constrained to be nonnegative if $\sqcup_i$ is $\geq$; y_i is constrained to be nonpositive if $\sqcup_i$ is $\leq$; y_i is free if $\sqcup_i$ is $=$.

The following table can help remember the above.

Primal (min)	Dual (max)
$\geq$ constraint	≥ 0 variable
$\leq$ constraint	≤ 0 variable
$=$ constraint	free variable
≥ 0 variable	$\leq$ constraint

Primal (min)	Dual (max)
≤ 0 variable	$\geq$ constraint
free variable	$=$ constraint

Below is an example of a primal-dual pair of problems based on the above definition:

Consider the primal problem:

$$\begin{array}{rrcrcrcl}
\min & x_1 & - & 2x_2 & + & 3x_3 & & \\
\text{s.t.} & -x_1 & & & + & 4x_3 & = & 5 \\
& 2x_1 & + & 3x_2 & - & 5x_3 & \geq & 6 \\
& & & 7x_2 & & & \leq & 8 \\
& x_1 & & & & & \geq & 0 \\
& & & x_2 & & & & \text{free} \\
& & & & & x_3 & \leq & 0.
\end{array}$$

Here, $\mathbf{A} = \begin{bmatrix} -1 & 0 & 4 \\ 2 & 3 & -5 \\ 0 & 7 & 0 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} 5 \\ 6 \\ 8 \end{bmatrix}$, and $\mathbf{c} = \begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix}$.

The primal problem has three constraints. So the dual problem has three variables. As the first constraint in the primal is an equation, the corresponding variable in the dual is free. As the second constraint in the primal is a $\geq$-inequality, the corresponding variable in the dual is nonnegative. As the third constraint in the primal is a $\leq$-inequality, the corresponding variable in the dual is nonpositive. Now, the primal problem has three variables. So the dual problem has three constraints. As the first variable in the primal is nonnegative, the corresponding constraint in the dual is a $\leq$-inequality. As the second variable in the primal is free, the corresponding constraint in the dual is an equation. As the third variable in the primal is nonpositive, the corresponding constraint in the dual is a $\geq$-inequality. Hence, the dual problem is:

$$\begin{array}{rrcrcrcl}
\max & 5y_1 & + & 6y_2 & + & 8y_3 & & \\
\text{s.t.} & -y_1 & + & 2y_2 & & & \leq & 1 \\
& & & 3y_2 & + & 7y_3 & = & -2 \\
& 4y_1 & - & 5y_2 & & & \geq & 3 \\
& y_1 & & & & & & \text{free} \\
& & & y_2 & & & \geq & 0 \\
& & & & & y_3 & \leq & 0.
\end{array}$$

Remarks. Note that in some books, the primal problem is always a maximization problem. In that case, what is our primal problem is their dual problem and what is our dual problem is their primal problem.

One can now prove a more general version of Theorem **??** as stated below. The details are left as an exercise.

Theorem 3.6: Duality Theorem for Linear Programming

Let (P) and (D) denote a primal-dual pair of linear programming problems. If either (P) or (D) has an optimal solution, then so does the other. Furthermore, the optimal values of the two problems are equal.

Theorem **??** is also known informally as **strong duality**.

Exercises

1. Write down the dual problem of

$$\begin{array}{rrcrcl} \min & 4x_1 & - & 2x_2 & & \\ \text{s.t.} & x_1 & + & 2x_2 & \geq & 3 \\ & 3x_1 & - & 4x_2 & = & 0 \\ & & & x_2 & \geq & 0. \end{array}$$

2. Write down the dual problem of the following:

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

3. Write down the dual problem of the following:

$$\begin{array}{rrcrcrcl} \min & x_1 & & & - & 9x_3 & & \\ \text{s.t.} & x_1 & - & 3x_2 & + & 2x_3 & = & 1 \\ & x_1 & & & & & \leq & 0 \\ & & & x_2 & & & & \text{free} \\ & & & & & x_3 & \geq & 0. \end{array}$$

4. Determine all values c_1, c_2 such that the linear programming problem

$$\begin{array}{rl} \min & c_1x_1 + c_2x_2 \\ \text{s.t.} & 2x_1 + x_2 \geq 2 \\ & x_1 + 3x_2 \geq 1. \end{array}$$

has an optimal solution. Justify your answer

Solutions

1. The dual is

$$\begin{array}{rrcrcr}
\max & 3y_1 & & & & \\
\text{s.t.} & y_1 & + & 3y_2 & = & 4 \\
 & 2y_1 & - & 4y_2 & \leq & -2 \\
 & y_1 & & & \geq & 0.
\end{array}$$

2. The dual is

$$\begin{array}{rrcrcl}
\max & y_1 & & & & \\
\text{s.t.} & y_1 & + & y_2 & \leq & 0 \\
 & y_1 & & & \leq & 3 \\
 & 2y_1 & - & 3y_2 & \leq & 1 \\
 & y_1 & & & & \text{free} \\
 & & & y_2 & \leq & 0.
\end{array}$$

3. The dual is

$$\begin{array}{rrcl}
\max & y_1 & & \\
\text{s.t.} & y_1 & \geq & 1 \\
 & -3y_1 & = & 0 \\
 & 2y_1 & \leq & -9 \\
 & y_1 & & \text{free.}
\end{array}$$

4. Let (P) denote the given linear programming problem.

   Note that $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ is a feasible solution to (P). Therefore, by Theorem **??**, it suffices to find all values c_1, c_2 such that

   (P) is not unbounded. This amounts to finding all values c_1, c_2 such that the dual problem of (P) has a feasible solution.

   The dual problem of (P) is

$$\begin{array}{rl}
\max & 2y_1 + y_2 \\
 & 2y_1 + y_2 = c_1 \\
 & y_1 + 3y_2 = c_2 \\
 & y_1, y_2 \geq 0.
\end{array}$$

   The two equality constraints gives $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{3}{5}c_1 - \frac{1}{5}c_2 \\ -\frac{1}{5}c_1 + \frac{2}{5}c_2 \end{bmatrix}$. Thus, the dual problem is feasible if and only if c_1 and c_2 are real numbers satisfying

$$\begin{aligned}
\frac{3}{5}c_1 - \frac{1}{5}c_2 \geq & \quad 0 \\
-\frac{1}{5}c_1 + \frac{2}{5}c_2 \geq & \quad 0,
\end{aligned}$$

or more simply,

$$\frac{1}{3}c_2 \leq c_1 \leq 2c_2.$$

3.6 Complementary slackness

Theorem 3.7: Weak Duality

Let (P) and (D) denote a primal-dual pair of linear programming problems in generic form as defined previously. Let $\mathbf{x}^*$ be a feasible solution to (P) and $\mathbf{y}^*$ is a feasible solution to (D). Then the following hold:

1. $\mathbf{c}^\mathsf{T}\mathbf{x}^* \geq \mathbf{y}^{*\mathsf{T}}\mathbf{b}$.
2. $\mathbf{x}^*$ *and* $\mathbf{y}^*$ *are optimal solutions to the respective problems if and only if the following conditions (known as the* ***complementary slackness conditions****) hold:*

$$\begin{array}{lll} x_j^* = 0 & \text{or} \quad \mathbf{y}^{*\mathsf{T}}\mathbf{A}_j = c_j & \text{for } j = 1, \ldots, n \\ y_i^* = 0 & \text{or} \quad {\mathbf{a}^{(i)}}^\mathsf{T}\mathbf{x}^* = b_i & \text{for } i = 1, \ldots, m \end{array}$$

Part 1 of the theorem is known as **weak duality**. Part 2 of the theorem is often called the **Complementary Slackness Theorem**.

Proof. [*Proof* of Theorem **??**]

Note that if x_j^* is constrained to be nonnegative, its corresponding dual constraint is $\mathbf{y}^{*\mathsf{T}}\mathbf{A}_j \leq c_j$. Hence, $(c_j - \mathbf{y}^{*\mathsf{T}}\mathbf{A}_j)x_j^* \geq 0$ with equality if and only if $x_j^* = 0$ or $\mathbf{y}^{*\mathsf{T}}\mathbf{A}_j = c_j$ (or both).

If x_j^* is constrained to be nonpositive, its corresponding dual constraint is $\mathbf{y}^{*\mathsf{T}}\mathbf{A}_j \geq c_j$. Hence, $(c_j - \mathbf{y}^{*\mathsf{T}}\mathbf{A}_j)x_j^* \geq 0$ with equality if and only if $x_j^* = 0$ or $\mathbf{y}^{*\mathsf{T}}\mathbf{A}_j = c_j$ (or both).

If x_j^* is free, its corresponding dual constraint is $\mathbf{y}^{*\mathsf{T}}\mathbf{A}_j = c_j$. Hence, $(c_j - \mathbf{y}^{*\mathsf{T}}\mathbf{A}_j)x_j^* = 0$.

We can combine these three cases and obtain that $(\mathbf{c}^\mathsf{T} - \mathbf{y}^{*\mathsf{T}}\mathbf{A})\mathbf{x}^* = \sum_{j=1}^{n}(c_j - \mathbf{y}^{*\mathsf{T}}\mathbf{A}_j)x_j^* \geq 0$ with equality if and only if for each $j = 1, \ldots, n$,

$$x_j^* = 0 \text{ or } \mathbf{y}^{*\mathsf{T}}\mathbf{A}_j = c_j.$$

(Here, the usage of "or" is not exclusive.)

Similarly, $\mathbf{y}^{*\mathsf{T}}(\mathbf{A}\mathbf{x}^* - \mathbf{b}) = \sum_{i=1}^{n} y_i^*({\mathbf{a}^{(i)}}^\mathsf{T}\mathbf{x}^* - b_i) \geq 0$ with equality if and only if for each $i = 1, \ldots, n$,

$$y_i^* = 0 \text{ or } {\mathbf{a}^{(i)}}^\mathsf{T}\mathbf{x}^* = b_i.$$

(Again, the usage of "or" is not exclusive.)

Adding the inequalities $(\mathbf{c}^\mathsf{T} - \mathbf{y^*}^\mathsf{T}\mathbf{A})\mathbf{x}^* \geq 0$ and $\mathbf{y^*}^\mathsf{T}(\mathbf{A}\mathbf{x}^* - \mathbf{b}) \geq 0$, we obtain $\mathbf{c}^\mathsf{T}\mathbf{x}^* - \mathbf{y^*}^\mathsf{T}\mathbf{b} \geq 0$ with equality if and only if the complementary slackness conditions hold. By strong duality, $\mathbf{x}^*$ is optimal (P) and $\mathbf{y}^*$ is optimal for (D) if and only if $\mathbf{c}^\mathsf{T}\mathbf{x}^* = \mathbf{y^*}^\mathsf{T}\mathbf{b}$. The result now follows.

♠

The complementary slackness conditions give a characterization of optimality which can be useful in solving certain problems as illustrated by the following example.

Example 3.8: Checking Optimality

Let (P) denote the following linear programming problem:

$$\begin{array}{rrcrcrcl}
\textit{min} & 2x_1 & + & 4x_2 & + & 2x_3 & & \\
\textit{s.t.} & x_1 & + & x_2 & + & 3x_3 & \leq & 1 \\
 & -x_1 & + & 2x_2 & + & x_3 & \geq & 1 \\
 & & & 3x_2 & - & 6x_3 & = & 0 \\
 & x_1 & , & & & x_3 & \geq & 0 \\
 & & & x_2 & & & & \textit{free.}
\end{array}$$

Is $\mathbf{x}^* = \begin{bmatrix} 0 \\ \frac{2}{5} \\ \frac{1}{5} \end{bmatrix}$ *an optimal solution to (P)?*

Solution. One could answer this question by solving (P) and then see if the objective function value of $\mathbf{x}^*$, assuming that its feasibiilty has already been verified, is equal to the optimal value. However, there is a way to make use of the given information to save some work.

Let (D) denote the dual problem of (P):

$$\begin{array}{rrcrcrcl}
\max & y_1 & + & y_2 & & & & \\
\text{s.t.} & y_1 & - & y_2 & & & \leq & 2 \\
 & y_1 & + & 2y_2 & + & 3y_3 & = & 4 \\
 & 3y_1 & + & y_2 & - & 6y_3 & \leq & 2 \\
 & y_1 & & & & & \leq & 0 \\
 & & & y_2 & & & \geq & 0 \\
 & & & & & y_3 & & \text{free.}
\end{array}$$

One can check that $\mathbf{x}^*$ is a feasible solution to (P). If $\mathbf{x}^*$ is optimal, then there must exist a feasible solution $\mathbf{y}^*$ to (D) satisfying together with $\mathbf{x}^*$ the complementary slackness conditions:

$$\begin{array}{rcr}
y_1^* = 0 & \text{or} & x_1^* + x_2^* + 3x_3^* = 1 \\
y_2^* = 0 & \text{or} & -x_1^* + 2x_2^* + x_3^* = 1 \\
y_3^* = 0 & \text{or} & 3x_2^* - 6x_3^* = 0 \\
x_1^* = 0 & \text{or} & y_1^* - y_2^* = 2 \\
x_2^* = 0 & \text{or} & y_1^* + 2y_2^* + 3y_3^* = 4 \\
x_3^* = 0 & \text{or} & 3y_1^* + y_2^* - 6y_3^* = 2.
\end{array}$$

As $x_2^*, x_3^* > 0$, satisfying the above conditions require that

$$\begin{aligned} y_1^* + 2y_2^* + 3y_3^* &= 4 \\ 3y_1^* + y_2^* - 6y_3^* &= 2. \end{aligned}$$

Solving for y_2^* and y_3^* in terms of y_1^* gives $y_2^* = 2 - y_1^*$, $y_3^* = \frac{1}{3}y_1^*$. To make $\mathbf{y}^*$ feasible to (D), we can set $y_1^* = 0$ to obtain the feasible solution $y_1^* = 0, y_2^* = 2$, $y_3^* = 0$. We can check that this $\mathbf{y}^*$ satisfies the complementary slackness conditions with $\mathbf{x}^*$. Hence, $\mathbf{x}^*$ is an optimal solution to (P) by Theorem **??**, part 2. ♠

Exercises

1. Let (P) and (D) denote a primal-dual pair of linear programming problems. Prove that if (P) is not infeasible and (D) is infeasible, then (P) is unbounded.

2. Let (P) denote the following linear programming problem:

$$\begin{array}{rrcrcrcl} \min & & & 4x_2 & + & 2x_3 & & \\ \text{s.t.} & x_1 & + & x_2 & + & 3x_3 & \leq & 1 \\ & x_1 & - & 2x_2 & + & x_3 & \geq & 1 \\ & x_1 & + & 3x_2 & - & 6x_3 & = & 0 \\ & x_1 & , & & & x_3 & \geq & 0 \\ & & & x_2 & & & & \text{free.} \end{array}$$

Determine if $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \frac{3}{5} \\ -\frac{1}{5} \\ 0 \end{bmatrix}$ is an optimal solution to (P).

3. Let (P) denote the following linear programming problem:

$$\begin{array}{rrcrcrcl} \min & x_1 & + & 2x_2 & - & 3x_3 & & \\ \text{s.t.} & x_1 & + & 2x_2 & + & 2x_3 & = & 2 \\ & -x_1 & + & x_2 & + & x_3 & = & 1 \\ & -x_1 & + & x_2 & - & x_3 & \geq & 0 \\ & x_1 & , & x_2 & , & x_3 & \geq & 0 \end{array}$$

Determine if $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ is an optimal solution to (P).

4. Let m and n be positive integers. Let $\mathbf{A} \in \mathbb{R}^{m \times n}$. Let $\mathbf{b} \in \mathbb{R}^m$. Let $\mathbf{c} \in \mathbb{R}^n$. Let (P) denote the linear programming problem

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

Let (D) denote the dual problem of (P):

$$\begin{array}{rl} \max & \mathbf{y}^\mathsf{T}\mathbf{b} \\ \text{s.t.} & \mathbf{y}^\mathsf{T}\mathbf{A} \leq \mathbf{c}^\mathsf{T}. \end{array}$$

Suppose that $\mathbf{A}$ has rank m and that (P) has at least one optimal solution. Prove that if $x_j^* = 0$ for *every* optimal solution $\mathbf{x}^*$ to (P), then there exists an optimal solution $\mathbf{y}^*$ to (D) such that ${\mathbf{y}^*}^\mathsf{T}\mathbf{A}_j < c_i$ where $\mathbf{A}_j$ denotes the jth column of $\mathbf{A}$.

Solutions

1. By the Fundamental Theorem of Linear Programming, (P) either is unbounded or has an optimal solution. If it is the latter, then by strong duality, (D) has an optimal solution, which contradicts that (D) is infeasible. Hence, (P) must be unbounded.

2. We show that it is not an optimal solution to (P). First, note that the dual problem of (P) is

$$\begin{array}{rrcrcrcl} \max & y_1 & + & y_2 & & & & \\ \text{s.t.} & y_1 & + & y_2 & + & y_3 & \leq & 0 \\ & y_1 & - & 2y_2 & + & 3y_3 & = & 4 \\ & 3y_1 & + & y_2 & - & 6y_3 & \leq & 2 \\ & y_1 & & & & & \leq & 0 \\ & & & y_2 & & & \geq & 0 \\ & & & & & y_3 & & \text{free.} \end{array}$$

\end{bmatrix}) were an optimal solution, there would exist $\mathbf{y}^*$ feasible to (D) satisfying the complementary slackness conditions with $\mathbf{x}^*$:

$$\begin{array}{rcl} y_1^* = 0 & \text{or} & x_1^* + x_2^* + 3x_3^* = 1 \\ y_2^* = 0 & \text{or} & x_1^* - 2x_2^* + x_3^* = 1 \\ y_3^* = 0 & \text{or} & x_1^* + 3x_2^* - 6x_3^* = 0 \\ x_1^* = 0 & \text{or} & y_1^* + y_2^* + y_3^* = 0 \\ x_2^* = 0 & \text{or} & y_1^* - 2y_2^* + 3y_3^* = 4 \\ x_3^* = 0 & \text{or} & 3y_1^* + y_2^* - 6y_3^* = 2. \end{array}$$

Since $x_1^* + x_2^* + 3x_3^* < 1$, we must have $y_1^* = 0$. Also, x_1^*, x_2^* are both nonzero. Hence,

$$\begin{aligned} y_1^* + y_2^* + y_3^* &= 0 \\ y_1^* - 2y_2^* + 3y_3^* &= 4, \end{aligned}$$

implying that

$$\begin{aligned} y_2^* + y_3^* &= 0 \\ -2y_2^* + 3y_3^* &= 4. \end{aligned}$$

Solving gives $y_2^* = -\frac{4}{5}$ and $y_3^* = \frac{4}{5}$. But this implies that y^* is not a feasible solution to the dual problem since we need $y_2^* \geq 0$. Hence, $\mathbf{x}^*$ is not an optimal solution to (P).

3. We show that it is not an optimal solution to (P). First, note that the dual problem of (P) is

$$\begin{array}{lrcrcrcr} \max & 2y_1 & + & y_2 & & & & \\ \text{s.t.} & y_1 & - & y_2 & - & y_3 & \leq & 1 \\ & 2y_1 & + & y_2 & + & y_3 & \leq & 2 \\ & 2y_1 & + & y_2 & - & y_3 & \leq & -3 \\ & y_1 & , & y_2 & & & & \text{free.} \\ & & & & & y_3 & \geq & 0 \end{array}$$

Note that $\mathbf{x}^* = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ is a feasible solution to (P). If it were an optimal solution to (P), there would exist $\mathbf{y}^*$ feasible to the dual problem (D) satisfying the complementary slackness conditions with $\mathbf{x}^*$:

$$\begin{array}{rcl} y_1^* = 0 & \text{or} & x_1^* + 2x_2^* + 2x_3^* = 2 \\ y_2^* = 0 & \text{or} & -x_1^* + x_2^* + x_3^* = 1 \\ y_3^* = 0 & \text{or} & -x_1^* + x_2^* - x_3^* = 0 \\ x_1^* = 0 & \text{or} & y_1^* - y_2^* - y_3^* = 1 \\ x_2^* = 0 & \text{or} & 2y_1^* + y_2^* + y_3^* = 2 \\ x_3^* = 0 & \text{or} & 2y_1^* + y_2^* - y_3^* = -3. \end{array}$$

Since $-x_1^* + x_2^* - x_3^* > 0$, we must have $y_3^* = 0$. Also, $x_2^* > 0$ implies that $2y_1^* + y_2^* + y_3^* = 2$. Simplifying gives $y_2^* = 2 - 2y_1^*$.

Hence, for y^* to be feasible to the dual problem, it needs to satisfy the third constraint, $2y_1^* + (2 - 2y_1^*) \leq -3$, which simplifies to the absurdity $2 \leq -3$. Hence, $\mathbf{x}^*$ is not an optimal solution to (P).

4. Let v denote the optimal value of (P). Let (P') denote the problem

$$\begin{array}{ll} \min & -x_i \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{c}^\mathsf{T}\mathbf{x} \leq v \\ & \mathbf{x} \geq 0 \end{array}$$

Note that x^* is a feasible solution to (P') if and only if it is an optimal solution to (P). Since $x_i^* = 0$ for every optimal solution to (P), we see that the optimal value of (P') is 0.

Let (D') denote the dual problem of (P'):

$$\begin{array}{rll} \max & \mathbf{y}^\mathsf{T}\mathbf{b} + vu & \\ \text{s.t.} & \mathbf{y}^\mathsf{T}\mathbf{A}_p + c_p u \leq 0 & \text{for all } p \neq i \\ & \mathbf{y}^\mathsf{T}\mathbf{A}_i + c_i u \leq -1 & \\ & u \leq 0. & \end{array}$$

Suppose that an optimal solution to (D') is given by $\mathbf{y}', u'$. Let $\bar{\mathbf{y}}$ be an optimal solution to (D). We consider two cases.

Case 1: $u' = 0$.

Then ${\mathbf{y}'}^\mathsf{T}\mathbf{b} = 0$. Hence, $\mathbf{y}^* = \bar{\mathbf{y}} + \mathbf{y}'$ is an optimal solution to (D) with ${\mathbf{y}^*}^\mathsf{T}\mathbf{A}_i < c_i$.

Case 2: $u' < 0$.

Then ${\mathbf{y}'}^\mathsf{T}\mathbf{b} + vu' = 0$, implying that $\frac{1}{|u'|}{\mathbf{y}'}^\mathsf{T}\mathbf{b} = v$. Let $\mathbf{y}^* = \frac{1}{|u|}\mathbf{y}'$. Then $\mathbf{y}^*$ is an optimal solution to (D) with ${\mathbf{y}^*}^\mathsf{T}\mathbf{A}_i < c_i$.

3.7 Basic feasible solution

For a linear constraint $\mathbf{a}^\mathsf{T}\mathbf{x} \sqcup \gamma$ where $\sqcup$ is $\geq$, $\leq$, or $=$, we call $\mathbf{a}^\mathsf{T}$ the **coefficient row-vector** of the constraint.

Let S denote a system of linear constraints with n variables and m constraints given by ${\mathbf{a}^{(i)}}^\mathsf{T}\mathbf{x} \sqcup_i b_i$ where $\sqcup_i$ is $\geq$, $\leq$, or $=$ for $i = 1, \ldots, m$.

For $\mathbf{x}' \in \mathbb{R}^n$, let $J(S, \mathbf{x}')$ denote the set $\{i \ : \ {\mathbf{a}^{(i)}}^\mathsf{T}\mathbf{x}' = b_i\}$ and define $\mathbf{A}_{S,\mathbf{x}'}$ to be the matrix whose rows are precisely the coefficient row-vectors of the constraints indexed by $J(S, \mathbf{x}')$.

Example 3.9

Suppose that S is the system

$$\begin{aligned} x_1 + x_2 - x_3 &\geq 2 \\ 3x_1 - x_2 + x_3 &= 2 \\ 2x_1 - x_2 &\leq 1 \end{aligned}$$

If $\mathbf{x}' = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix}$, *then* $J(S, \mathbf{x}') = \{1, 2\}$ *since* $\mathbf{x}'$ *satisfies the first two constraints with equality but not the* *third. Hence,* $\mathbf{A}_{S,\mathbf{x}'} = \begin{bmatrix} 1 & 1 & -1 \\ 3 & -1 & 1 \end{bmatrix}$.

Definition 3.10

A solution $\mathbf{x}^*$ *to* S *is called a* ***basic feasible solution*** *if the rank of* $\mathbf{A}_{S,\mathbf{x}^*}$ *is* n.

A basic feasible solution to the system in Example **??** is $\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$.

It is not difficult to see that in two dimensions, basic feasible solutions correspond to "corner points" of the set of all solutions. Therefore, the notion of a basic feasible solution generalizes the idea of a corner point to higher dimensions.

The following result is the basis for what is commonly known as the **corner method** for solving linear programming problems in two variables.

Theorem 3.11: Basic Feasible Optimal Solution

Let (P) be a linear programming problem. Suppose that (P) has an optimal solution and there exists a basic feasible solution to its constraints. Then there exists an optimal solution that is a basic feasible solution.

We first state the following simple fact from linear algebra:

Lemma 3.12

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ *and* $\mathbf{d} \in \mathbb{R}^n$ *be such that* $\mathbf{A}\mathbf{d} = 0$. *If* $\mathbf{q} \in \mathbb{R}^n$ *satisfies* $\mathbf{q}^\top \mathbf{d} \neq 0$ *then* $\mathbf{q}^T$ *is not in the row space of* $\mathbf{A}$.

Proof.

Proof of Theorem **??**.
Suppose that the system of constraints in (P), call it S, has m constraints and n variables. Let the objective function be $\mathbf{c}^\top \mathbf{x}$. Let v denote the optimal value.

Let $\mathbf{x}^*$ be an optimal solution to (P) such that the rank of $\mathbf{A}_{S,\mathbf{x}^*}$ is as large as possible. We claim that $\mathbf{x}^*$ must be a basic feasible solution.

To ease notation, let $J = J(S, \mathbf{x}^*)$. Let $N = \{1, \ldots, m\} \backslash J$.

Suppose to the contrary that the rank of $\mathbf{A}_{S,\mathbf{x}^*}$ is less than n. Let $\mathbf{P}\mathbf{x} = \mathbf{q}$ denote the system of equations obtained by setting the constraints indexed by J to equalities. Then $\mathbf{P}\mathbf{x} = \mathbf{A}_{S,\mathbf{x}^*}$. Since $\mathbf{P}$ has n columns and its rank is less than n, there exists a nonzero $\mathbf{d}$ such that $\mathbf{P}\mathbf{d} = 0$.

As $\mathbf{x}^*$ satisfies each constraint indexed by N strictly, for a sufficiently small $\varepsilon > 0$, $\mathbf{x}^* + \varepsilon\mathbf{d}$ and $\mathbf{x}^* - \varepsilon\mathbf{d}$ are solutions to S and therefore are feasible to (P). Thus,

$$\begin{aligned} \mathbf{c}^\mathsf{T}(\mathbf{x}^* + \varepsilon\mathbf{d}) &\geq v \\ \mathbf{c}^\mathsf{T}(\mathbf{x}^* - \varepsilon\mathbf{d}) &\geq v. \end{aligned} \tag{3.1}$$

Since $\mathbf{x}^*$ is an optimal solution, we have $\mathbf{c}^\mathsf{T}\mathbf{x}^* = v$. Hence, (**??**) simplifies to

$$\begin{aligned} \varepsilon\mathbf{c}^\mathsf{T}\mathbf{d} &\geq 0 \\ -\varepsilon\mathbf{c}^\mathsf{T}\mathbf{d} &\geq 0, \end{aligned}$$

giving us $\mathbf{c}^\mathsf{T}\mathbf{d} = 0$ since $\varepsilon > 0$.

Without loss of generality, assume that the constraints indexed by N are $\mathbf{Q}\mathbf{x} \geq \mathbf{r}$. As (P) does have a basic feasible solution, implying that the rank of $\begin{bmatrix}\mathbf{P}\\\mathbf{Q}\end{bmatrix}$ is n, at least one row of $\mathbf{Q}$, which we denote by $\mathbf{t}^\mathsf{T}$, must satisfy $\mathbf{t}^\mathsf{T}\mathbf{d} \neq 0$. Without loss of generality, we may assume that $\mathbf{t}^\mathsf{T}\mathbf{d} > 0$, replacing $\mathbf{d}$ with $-\mathbf{d}$ if necessary.

Consider the linear programming problem

$$\begin{array}{rl} \min & \lambda \\ \text{s.t.} & \mathbf{Q}(\mathbf{x}^* + \lambda\mathbf{d}) \geq \mathbf{p} \end{array}$$

Since at least one entry of $\mathbf{Q}\mathbf{d}$ is positive (namely, $\mathbf{t}^\mathsf{T}\mathbf{d}$), this problem must have an optimal solution, say λ'. Setting $\mathbf{x}' = \mathbf{x}^* + \lambda'$, we have that $\mathbf{x}'$ is an optimal solution since $\mathbf{c}^\mathsf{T}\mathbf{x}' = v$.

Now, $\mathbf{x}'$ must satisfy at least one constraint in $\mathbf{Q} \geq \mathbf{p}$ with equality. Let $\mathbf{q}^\mathsf{T}$ be the coefficient row-vector of one such constraint. Then the rows of $\mathbf{A}_{S,\mathbf{x}'}$ must have all the rows of $\mathbf{A}_{S,\mathbf{x}^*}$ and $\mathbf{q}^\mathsf{T}$. Since $\mathbf{q}^\mathsf{T}\mathbf{d} \neq 0$, by Lemma **??**, the rank of $\mathbf{A}_{S,\mathbf{x}'}$ is larger than rank the rank of $\mathbf{A}_{S,\mathbf{x}^*}$, contradicting our choice of $\mathbf{x}^*$. Thus, $\mathbf{x}^*$ must be a basic feasible solution. ♠

Exercises

1. Find all basic feasible solutions to

$$\begin{aligned} x_1 + 2x_2 - x_3 &\geq 1 \\ x_2 + 2x_3 &\geq 3 \\ -x_1 + 2x_2 + x_3 &\geq 3 \\ -x_1 + x_2 + x_3 &\geq 0. \end{aligned}$$

2. A set $S \subset \mathbb{R}^n$ is said to be bounded if there exists a real number $M > 0$ such that for every $\mathbf{x} \in S$, $|x_i| < M$ for all $i = 1, \ldots, n$. Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$. Prove that if $\{\mathbf{x} : \mathbf{A}\mathbf{x} \geq \mathbf{b}\}$ is nonempty and bounded, then there is a basic feasible solution to $\mathbf{A}\mathbf{x} \geq \mathbf{b}$.

3. Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$ where m and n are positive integers with $m \leq n$. Suppose that the rank of $\mathbf{A}$ is m and $\mathbf{x}'$ is a basic feasible solution to

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &\geq 0. \end{aligned}$$

Let $J = \{i \,:\, x'_i > 0\}$. Prove that the columns of $\mathbf{A}$ indexed by J are linearly independent.

Solutions

1. To obtain all the basic feasible solutions, it suffices to enumerate all subsystems $\mathbf{A}'\mathbf{x} \geq \mathbf{b}'$ of the given system such that the rank of $\mathbf{A}'$ is three and solve $\mathbf{A}'\mathbf{x} = \mathbf{b}'$ for $\mathbf{x}$ and see if is a solution to the system, in which case it is a basic feasible solution. Observe that every basic feasible solution can be discovered in this manner.

   We have at most four subsystems to consider.

   Setting the first three inequalities to equality gives the unique solution $\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ which satisfies the given system.. Hence, $\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ is a basic feasible solution.

   Setting the first, second, and fourth inequalities to equality gives the unique solution $\begin{bmatrix} \frac{5}{3} \\ \frac{1}{3} \\ \frac{4}{3} \end{bmatrix}$ which violates the third inequality of the given system.

   Setting the first, third, and fourth inequalities to equality leads to no solution. (In fact, the coefficient matrix of the system does not have rank 3 and therefore this case can be ignored.)

   Setting the last three inequalities to equality gives the unique solution $\begin{bmatrix} 3 \\ 3 \\ 0 \end{bmatrix}$ which satisfies the given system. Hence, $\begin{bmatrix} 3 \\ 3 \\ 0 \end{bmatrix}$ is a basic feasible solution.

   Thus, $\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ and $\begin{bmatrix} 3 \\ 3 \\ 0 \end{bmatrix}$ are the only basic feasible solutions.

2. Let S denote the system $\mathbf{A}\mathbf{x} \geq \mathbf{b}$. Let $\mathbf{x}'$ be a solution to S such that the rank of $\mathbf{A}_{S,\mathbf{x}'}$ is as large as possible. If the rank is n, then we are done. Otherwise, there exists nonzero $\mathbf{d} \in \mathbb{R}^n$ such $\mathbf{A}_{S,\mathbf{x}'}\mathbf{d} = 0$. Since the set of solutions to S is a bounded set, at least one of the following values is finite:

- $\max\{\lambda : \mathbf{A}(\mathbf{x}' + \lambda\mathbf{d}) \geq \mathbf{b}\}$
- $\min\{\lambda : \mathbf{A}(\mathbf{x}' + \lambda\mathbf{d}) \geq \mathbf{b}\}$

Without loss of generality, assume that the maximum is finite and is equal to λ^*. Setting $\mathbf{x}^*$ to $\mathbf{x}' + \lambda^*\mathbf{d}$, we have that the rows of $\mathbf{A}_{S,\mathbf{x}^*}$ contains all the rows of $\mathbf{A}_{S,\mathbf{x}'}$ plus at least one additional row, say $\mathbf{q}^\mathsf{T}$. Since $\mathbf{q}^\mathsf{T}\mathbf{d} \neq 0$, by Lemma **??**, the rank of $\mathbf{A}_{S,\mathbf{x}^*}$ is larger than the rank of $\mathbf{A}_{S,\mathbf{x}'}$, contradicting our choice of $\mathbf{x}'$.

3. The system of equations obtained from taking all the constraints satisfied with equality by $\mathbf{x}'$ is

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \mathbf{b} \\ x_j &= 0 \quad j \notin J. \end{aligned} \tag{3.2}$$

Note that the coefficient matrix of this system has rank n if and only if it has a unique solution. Now, (**??**) simplifies to

$$\sum_{j \in J} x_j \mathbf{A}_j = \mathbf{b},$$

which has a unique solution if and only if the columns of $\mathbf{A}$ indexed by J are linearly independent.

4. LP Notes from ISE 5405

4.1 Introduction to Optimization

Optimization (i.e., Mathematical Programming) seeks to select, from a set of alternative solutions (decisions), a solution that is “best" for a given performance criteria (i.e., maximize or minimizes the criteria). The following is a general optimization problem:

$$\max\{f(\mathbf{x},\mathbf{y}) : A(\mathbf{x}) + G(\mathbf{y}) \leq \mathbf{b}_1, H(\mathbf{x}) + W(\mathbf{y}) = \mathbf{b}_2, \mathbf{x} \in \mathscr{Z}^+, \mathbf{y} \in \mathscr{R}^+\},$$

where $\mathbf{x}$ and $\mathbf{y}$ are vectors of decision variables, $f(\mathbf{x},\mathbf{y})$ is the *objective function*, which defines the “best” solution (in this case the optimization problem seeks to maximize the objective function), and $A(\mathbf{x}) + G(\mathbf{y}) \leq \mathbf{b}_1$, $H(\mathbf{x}) + W(\mathbf{y}) = \mathbf{b}_2$, $\mathbf{x} \in \mathscr{Z}^+$, and $\mathbf{y} \in \mathscr{R}^+$ are the *constraints* that define the set of possible solutions.

Depending on the nature of the objective function, the constraints, and the input parameters, we can make some broad classifications of optimization problems, as follows:

Linear Optimization: Linear optimization, i.e., a linear program (LP), has a linear objective function subject to a set of linear constraints and continuous decision variables.

Definition: A function $f(x_1,x_2,\cdots,x_n)$ is linear if, and only if, we have $f(x_1,x_2,\cdots,x_n) = c_1x_1 + c_2x_2 + \cdots + c_nx_n$, where the $c_1,c_2,\cdots,c_n$ coefficients are constants.

An LP has the following general form:

$$\max\{\mathbf{cx} : \mathbf{Ax} = \mathbf{b}, x \in \mathscr{R}\},$$

where $\mathbf{x}$ is a vector of decision variables, and the vectors $\mathbf{c}$ and $\mathbf{b}$, as well as the matrix $\mathbf{A}$, are constant problem parameters.

Nonlinear Optimization: Nonlinear optimization, i.e., a nonlinear program, is similar to an LP, but objective function and/or the constraints are nonlinear.

Integer Optimization: Integer optimization, , i.e., an integer program (IP), is much like an LP, but some) variables restricted to take only integer values.

To use optimization, first you must formulate your model, based on the system of interest and any simplifications required (i.e., assumptions). Formulating the model is not enough, we are also interested in solving the problem, and in a reasonable amount of time (however that is determined). To solve these problems, algorithms are developed. An algorithms is a step-by-step process for finding a solution. We can broadly define different types of algorithms as follows:

- Optimal algorithms - processes that solve the model to optimality (and proves optimality).
- Near-optimal algorithms with bounds (heuristics), processes that do not guarantee optimality, but provides “good" solutions with known bounds.
- Other heuristic algorithms, processes that provide a “good" solution, but bounds are not provided, or are not that useful.

Algorithms can also be categorized based on performance, for instance, usually a *polynomial time algorithm* is better than an *exponential time algorithm*.

The class will almost exclusively focus on Linear Programs (LP) because: 1) LP are useful for many problems; 2) LPs are, relatively, easy to solve; and most importantly 3) LP are an important foundation for further courses in optimization.

4.1.1. Notation

- We use bold text to indicate a matrix or vector, e.g., the matrix **A** or the vector **x**.

4.2 Linear Optimization

In this section, we study on linear optimization problems, i.e., linear programs (LPs).

4.2.1. Problem Formulation

Remember, for a linear program (LP), we want to maximize or minimize a linear **objective function** of the continous decision variables, while considering linear constraints on the values of the decision variables.

Definition 4.1: Linear Function

function $f(x_1, x_2, \cdots, x_n)$ *is linear if, and only if, we have* $f(x_1, x_2, \cdots, x_n) = c_1x_1 + c_2x_2 + \cdots + c_nx_n$, *where the* $c_1, c_2, \cdots, c_n$ *coefficients are constants.*

A Generic Linear Program (LP)

Decision Variables:
x_i : continuous variables ($x_i \in \mathscr{R}$, i.e., a real number), $\forall i = 1, \cdots, 3$.

Parameters (known input parameters):
c_i : cost coefficients $\forall i = 1, \cdots, 3$
a_{ij} : constraint coefficients $\forall i = 1, \cdots, 3,\ j = 1, \cdots, 4$
b_j : right hand side coefficient for constraint j, $j = 1, \cdots, 4$

$$\begin{aligned}
\text{Min}\quad & z = c_1x_1 + c_2x_2 + c_3x_3 && (4.1)\\
\text{s.t.}\quad & a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \geq b_1 && (4.2)\\
& a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \leq b_2 && (4.3)\\
& a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 && (4.4)\\
& a_{41}x_1 + a_{42}x_2 + a_{43}x_3 \geq b_4 && (4.5)\\
& x_1 \geq 0, x_2 \leq 0, x_3\ urs. && (4.6)
\end{aligned}$$

Eq. (**??**) is the objective function, (**??**)-(**??**) are the functional constraints, while (**??**) is the sign restrictions (*urs* signifies that the variable is unrestricted). If we were to add any one of these following constraints $x_2 \in \{0,1\}$ (x_2 is binary-valued) or $x_3 \in \mathscr{Z}$ (x_3 is integer-valued) we would have an Integer Program. For the purposes of this class, an Integer Program (IP) is just an LP with added integer restrictions on (some) variables.

While, in general, solvers will take any form of the LP, there are some special forms we use in analysis:

<u>LP Standard Form</u>: The standard form has all constraints as equalities, and all variables as non-negative. The generic LP is not in standard form, but any LP can be converted to standard form.

Since x_2 is non-positive and x_3 unrestricted, perform the following substitutions $x_2 = -\hat{x}_2$ and $x_3 = x_3^+ - x_3^-$, where $\hat{x}_2, x_3^+, x_3^- \geq 0$. Eqs. (**??**) and (**??**) are in the form left-hand side (LHS) $\geq$ right-hand side (RHS), so to make an equality, subtract a non-negative slack variable from the LHS (s_1 and s_4). Eq. (**??**) is in the form LHS $\leq$ RHS, so add a non-negative slack variable to the LHS.

$$\begin{aligned}
\text{Min } \; & z = c_1x_1 - c_2\hat{x}_2 + c_3(x_3^+ - x_3^-) \\
\text{s.t. } \; & a_{11}x_1 - a_{12}x_2 + a_{13}(x_3^+ - x_3^-) - s_1 = b_1 \\
& a_{21}x_1 - a_{22}\hat{x}_2 + a_{23}(x_3^+ - x_3^-) + s_2 = b_2 \\
& a_{31}x_1 - a_{32}\hat{x}_2 + a_{33}(x_3^+ - x_3^-) = b_3 \\
& a_{41}x_1 - a_{42}\hat{x}_2 + a_{43}x_3 - s_4 = b_4 \\
& x_1, \hat{x}_2, x_3^+, x_3^-, s_1, s_2, s_4 \geq 0.
\end{aligned}$$

<u>LP Canonical Form</u>: For a minimization problem the canonical form of the LP has the LHS of each constraint greater than or equal to the the RHS, and a maximization the LHS less than or equal to the RHS, and non-negative variables.

Next we consider some formulation examples:

Production Problem: You have 21 units of transparent aluminum alloy (TAA), LazWeld1, a joining robot leased for 23 hours, and CrumCut1, a cutting robot leased for 17 hours of aluminum cutting. You also have production code for a bookcase, desk, and cabinet, along with commitments to buy any of these you can produce for \$18, \$16, and \$10 apiece, respectively. A bookcase requires 2 units of TAA, 3 hours of joining, and 1 hour of cutting, a desk requires 2 units of TAA, 2 hours of joining, and 2 hour of cutting, and a cabinet requires 1 unit of TAA, 2 hours of joining, and 1 hour of cutting. Formulate an LP to maximize your revenue given your current resources.

<u>Decision variables:</u>
x_i : number of units of product i to produce,
$\forall i = \{bookcase,\ desk,\ cabinet\}$.

$$\begin{aligned}
\max z = 18x_1 + 16x_2 + 10x_3 : & \\
2x_1 + 2x_2 + 1x_3 \leq 21 & \qquad (TAA) \\
3x_1 + 2x_2 + 2x_3 \leq 23 & \qquad (LazWeld1) \\
1x_1 + 2x_2 + 1x_3 \leq 17 & \qquad (CrumCut1) \\
x_1, x_2, x_3 \geq 0. &
\end{aligned}$$

Work Scheduling Problem: You are the manager of LP Burger. The following table shows the minimum number of employees required to staff the restaurant on each day of the week. Each employees must work

Day of Week	Workers Required
1 = Monday	6
2 = Tuesday	4
3 = Wednesday	5
4 = Thursday	4
5 = Friday	3
6 = Saturday	7
7 = Sunday	7

for five consecutive days. Formulate an LP to find the minimum number of employees required to staff the restaurant.

Decision variables:
x_i : the number of workers that start 5 consecutive days of work on day i, $i = 1, \cdots, 7$

$$
\begin{aligned}
\text{Min} \quad & z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \\
\text{s.t.} \quad & x_1 + x_4 + x_5 + x_6 + x_7 \geq 6 \\
& x_2 + x_5 + x_6 + x_7 + x_1 \geq 4 \\
& x_3 + x_6 + x_7 + x_1 + x_2 \geq 5 \\
& x_4 + x_7 + x_1 + x_2 + x_3 \geq 4 \\
& x_5 + x_1 + x_2 + x_3 + x_4 \geq 3 \\
& x_6 + x_2 + x_3 + x_4 + x_5 \geq 7 \\
& x_7 + x_3 + x_4 + x_5 + x_6 \geq 7 \\
& x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0.
\end{aligned}
$$

The solution is as follows:

LP Solution	IP Solution
$z_{LP} = 7.333$	$z_I = 8.0$
$x_1 = 0$	$x_1 = 0$
$x_2 = 0.333$	$x_2 = 0$
$x_3 = 1$	$x_3 = 0$
$x_4 = 2.333$	$x_4 = 3$
$x_5 = 0$	$x_5 = 0$
$x_6 = 3.333$	$x_6 = 4$
$x_7 = 0.333$	$x_7 = 1$

LP Burger has changed it's policy, and allows, at most, two part time workers, who work for two consecutive days in a week. Formulate this problem.

Decision variables:
x_i : the number of workers that start 5 consecutive days of work on day i, $i = 1, \cdots, 7$
y_i : the number of workers that start 2 consecutive days of work on day i, $i = 1, \cdots, 7$.

$$
\begin{aligned}
\text{Min} \quad & z = 5(x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7) \\
& + 2(y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7) \\
\text{s.t.} \quad & x_1 + x_4 + x_5 + x_6 + x_7 + y_1 + y_7 \geq 6 \\
& x_2 + x_5 + x_6 + x_7 + x_1 + y_2 + y_1 \geq 4 \\
& x_3 + x_6 + x_7 + x_1 + x_2 + y_3 + y_2 \geq 5 \\
& x_4 + x_7 + x_1 + x_2 + x_3 + y_4 + y_3 \geq 4 \\
& x_5 + x_1 + x_2 + x_3 + x_4 + y_5 + y_4 \geq 3 \\
& x_6 + x_2 + x_3 + x_4 + x_5 + y_6 + y_5 \geq 7 \\
& x_7 + x_3 + x_4 + x_5 + x_6 + y_7 + y_6 \geq 7 \\
& y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 \leq 2 \\
& x_i \geq 0, y_i \geq 0, \forall i = 1, \cdots, 7.
\end{aligned}
$$

The Diet Problem: In the future (as envisioned in a bad 70's science fiction film) all food is in tablet form, and there are four types, green, blue, yellow, and red. A balanced, futuristic diet requires, at least 20 units of Iron, 25 units of Vitamin B, 30 units of Vitamin C, and 15 units of Vitamin D. Formulate an LP that ensures a balanced diet at the minimum possible cost.

Tablet	Iron	B	C	D	Cost ($)
green (1)	6	6	7	4	1.25
blue (2)	4	5	4	9	1.05
yellow (3)	5	2	5	6	0.85
red (4)	3	6	3	2	0.65

Now we formulate the problem:

Decision variables:
x_i : number of tablet of type i to include in the diet, $\forall i \in \{1, 2, 3, 4\}$.

$$
\begin{aligned}
\text{Min} \quad & z = 1.25x_1 + 1.05x_2 + 0.85x_3 + 0.65x_4 \\
\text{s.t.} \quad & 6x_1 + 4x_2 + 5x_3 + 3x_4 \geq 20 \\
& 6x_1 + 5x_2 + 2x_3 + 6x_4 \geq 25 \\
& 7x_1 + 4x_2 + 5x_3 + 3x_4 \geq 30 \\
& 4x_1 + 9x_2 + 6x_3 + 2x_4 \geq 15 \\
& x_1, x_2, x_3, x_4 \geq 0.
\end{aligned}
$$

The Next Diet Problem: Progress is important, and our last problem had too many tablets, so we are going to produce a single, purple, 10 gram tablet for our futuristic diet requires, which are at least 20 units of Iron, 25 units of Vitamin B, 30 units of Vitamin C, and 15 units of Vitamin D, and 2000 calories. The tablet is made from blending 4 nutritious chemicals; the following table shows the units of our nutrients per, and cost of, grams of each chemical. Formulate an LP that ensures a balanced diet at the minimum possible cost.

Tablet	Iron	B	C	D	Calories	Cost ($)
Chem 1	6	6	7	4	1000	1.25
Chem 2	4	5	4	9	250	1.05
Chem 3	5	2	5	6	850	0.85
Chem 4	3	6	3	2	750	0.65

Decision variables:
x_i : grams of chemical i to include in the purple tablet, $\forall i = 1,2,3,4$.

$$\begin{aligned}
\text{Min} z &= 1.25x_1 + 1.05x_2 + 0.85x_3 + 0.65x_4 \\
\text{s.t. } & 6x_1 + 4x_2 + 5x_3 + 3x_4 \geq 20 \\
& 6x_1 + 5x_2 + 2x_3 + 6x_4 \geq 25 \\
& 7x_1 + 4x_2 + 5x_3 + 3x_4 \geq 30 \\
& 4x_1 + 9x_2 + 6x_3 + 2x_4 \geq 15 \\
& 1000x_1 + 250x_2 + 850x_3 + 750x_4 \geq 2000 \\
& x_1 + x_2 + x_3 + x_4 = 10 \\
& x_1, x_2, x_3, x_4 \geq 0.
\end{aligned}$$

The Assignment Problem: Consider the assignment of n teams to n projects, where each team ranks the projects, where their favorite project is given a rank of n, their next favorite $n-1$, and their least favorite project is given a rank of 1. The assignment problem is formulated as follows (we denote ranks using the R-parameter):

Variables:
x_{ij} : 1 if project i assigned to team j, else 0.

$$\begin{aligned}
\text{Max } z &= \sum_{i=1}^{n} \sum_{j=1}^{n} R_{ij} x_{ij} \\
\text{s.t. } & \sum_{i=1}^{n} x_{ij} = 1, \ \ \forall j = 1, \cdots, n \\
& \sum_{j=1}^{n} x_{ij} = 1, \ \ \forall i = 1, \cdots, n \\
& x_{ij} \geq 0, \ \ \forall i = 1, \cdots, n, j = 1, \cdots, n.
\end{aligned}$$

The assignment problem has an integrality property, such that if we remove the binary restriction on the x variables (now just non-negative, i.e., $x_{ij} \geq 0$) then we still get binary assignments, despite the fact that it is now an LP. This property is very interesting and useful. Of course, the objective function might not quite what we want, we might be interested ensuring that the team with the worst assignment is as good as possible (a fairness criteria). One way of doing this is to modify the assignment problem using a max-min objective:

Max-min Assignment-like Formulation

$$\begin{aligned} Max \quad & z \\ s.t. \quad & \sum_{i=1}^{n} x_{ij} = 1, \;\; \forall j = 1, \cdots, n \\ & \sum_{j=1}^{n} x_{ij} = 1, \;\; \forall i = 1, \cdots, n \\ & x_{ij} \geq 0, \;\; \forall i = 1, \cdots, n, J = 1, \cdots, n \\ & z \leq \sum_{i=1}^{n} R_{ij} x_{ij}, \;\; \forall j = 1, \cdots, n. \end{aligned}$$

Does this formulation have the integrality property (it is not an assignment problem)? Consider a very simple example where two teams are to be assigned to two projects and the teams give the projects the following rankings: Both teams prefer Project 2. For both problems, if we remove the binary restriction on

	Project 1	Project 2
Team 1	2	1
Team 2	2	1

the x-variable, they can take values between (and including) zero and one. For the assignment problem the optimal solution will have $z = 3$, and fractional x-values will not improve z. For the max-min assignment problem this is not the case, the optimal solution will have $z = 1.5$, which occurs when each team is assigned half of each project (i.e., for Team 1 we have $x_{11} = 0.5$ and $x_{21} = 0.5$).

Linear Data Models: Consider a data set that consists of n data points (x_i, y_i). We want to fit the best line to this data, such that given an x-value, we can predict the associated y-value. Thus, the form is $y_i = \alpha x_i + \beta$ and we want to choose the α and β values such that we minimize the error for our n data points.

<u>**Variables:**</u>
e_i : error for data point i, $i = 1, \cdots, n$.
α : slope of fitted line.
β : intercept of fitted line.

$$
\begin{aligned}
Min \quad & \sum_{i=1}^{n} |e_i| \\
s.t. \quad & \alpha x_i + \beta - y_i = e_i, \;\; i = 1, \cdots, n \\
& e_i, \alpha, \beta \; urs.
\end{aligned}
$$

Of course, absolute values are not linear function, so we can linearize as follows:

<u>**Decision variables:**</u>
e_i^+ : positive error for data point i, $i = 1, \cdots, n$.
e_i^- : negative error for data point i, $i = 1, \cdots, n$.
α : slope of fitted line.
β : intercept of fitted line.

$$
\begin{aligned}
Min \quad & \sum_{i=1}^{n} e_i^+ + e_i^- \\
s.t. \quad & \alpha x_i + \beta - y_i = e_i^+ - e_i^-, \;\; i = 1, \cdots, n \\
& e_i^+, e_i^- \geq 0, \alpha, \beta \; urs.
\end{aligned}
$$

Two-Person Zero-Sum Games: Consider a game with two players, $\mathscr{A}$ and $\mathscr{B}$. In each round of the game, $\mathscr{A}$ chooses one out of m possible actions, while $\mathscr{B}$ chooses one out of n actions. If $\mathscr{A}$ takes action j while $\mathscr{B}$ takes action i, then c_{ij} is the payoff for $\mathscr{A}$, if $c_{ij} > 0$, $\mathscr{A}$ “wins" c_{ij} (and $\mathscr{B}$ losses that amount), and if $c_{ij} < 0$ if $\mathscr{B}$ “wins" $-c_{ij}$ (and $\mathscr{A}$ losses that amount). This is a two-person zero-sum game.

Rock, Paper, Scissors is a two-person zero-sum game, with the following payoff matrix.

		$\mathscr{A}$		
		R	**P**	**S**
	R	0	1	-1
$\mathscr{B}$	**P**	-1	0	1
	S	1	-1	0

We can have a similar game, but with a different payoff matrix, as follows:

		$\mathscr{A}$		
		R	**P**	**S**
	R	4	-1	-1
$\mathscr{B}$	**P**	-2	4	-2
	S	-3	-3	4

What is the optimal strategy for $\mathscr{A}$ (for either game)? We define x_j as the probability that $\mathscr{A}$ takes action j (related to the columns). Then the payoff for $\mathscr{A}$, if $\mathscr{B}$ takes action i is $\sum_{j=1}^{m} c_{ij}x_j$. Of course, $\mathscr{A}$ does not know what action $\mathscr{B}$ will take, so let's find a strategy that maximizes the minimum expected winnings of $\mathscr{A}$ given any random strategy of $\mathscr{B}$, which we can formulate as follows:

$$\begin{aligned} Max \quad & (min_{i=1,\cdots,n} \sum_{j=1}^{m} c_{ij}x_i) \\ s.t. \quad & \sum_{j=1}^{m} x_j = 1 \\ & x_j \geq 0, \;\; i = 1, \cdots, m, \end{aligned}$$

which can be linearized as follows:

$$\begin{aligned} Max \quad & z \\ s.t. \quad & z \leq \sum_{j=1}^{m} c_{ij}x_j, \;\; i = 1, \cdots, n \\ & \sum_{j=1}^{m} x_j = 1 \\ & x_j \geq 0, \;\; i = 1, \cdots, m. \end{aligned}$$

The last two constraints ensure the that x_i-variables are valid probabilities. If you solved this LP for the first game (i.e., payoff matrix) you find the best strategy is $x_1 = 1/3$, $x_2 = 1/3$, and $x_3 = 1/3$ and there is no expected gain for player $\mathscr{A}$. For the second game, the best strategy is $x_1 = 23/107$, $x_2 = 37/107$, and $x_3 = 47/107$, with $\mathscr{A}$ gaining, on average, $8/107$ per round.

4.2.2. Linear Algebra Review

Vectors and Linear and Convex Combinations

Vectors: Vector **n** has n-elements and represents a point (or an arrow from the origin to the point, denoting a direction) in $\mathscr{R}^n$ space (Euclidean or real space). Vectors can be expressed as either row or column vectors.

Vector Addition: Two vectors of the same size can be added, componentwise, e.g., for vectors $\mathbf{a} = (2,3)$ and $\mathbf{b} = (3,2)$, $\mathbf{a}+\mathbf{b} = (2+3, 3+2) = (5,5)$.

Scalar Multiplication: A vector can be multiplied by a scalar k (constant) component-wise. If $k > 0$ then this does not change the direction represented by the vector, it just scales the vector.

Inner or Dot Product: Two vectors of the same size can be multiplied to produce a real number. For example, $\mathbf{ab} = 2*3+3*2 = 10$.

Linear Combination: The vector $\mathbf{b}$ is a **linear combination** of $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k$ if $\mathbf{b} = \sum_{i=1}^{k} \lambda_i \mathbf{a}_i$ for $\lambda_1, \lambda_2, \cdots, \lambda_k \in \mathscr{R}$. If $\lambda_1, \lambda_2, \cdots, \lambda_k \in \mathscr{R}_{\geq 0}$ then $\mathbf{b}$ is a *non-negative linear combination* of $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k$.

Convex Combination: The vector $\mathbf{b}$ is a **convex combination** of $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k$ if $\mathbf{b} = \sum_{i=1}^{k} \lambda_i \mathbf{a}_i$, for $\lambda_1, \lambda_2, \cdots, \lambda_k \in \mathscr{R}_{\geq 0}$ and $\sum_{i=1}^{k} \lambda_i = 1$. For example, any convex combination of two points will lie on the line segment between the points.

Linear Independence: Vectors $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k$ are *linearly independent* if the following linear combination $\sum_{i=1}^{k} \lambda_i \mathbf{a}_i = 0$ implies that $\lambda_i = 0,\ i = 1, 2, \cdots, k$. In $\mathscr{R}^2$ two vectors are only linearly dependent if they lie on the same line. Can you have three linearly independent vectors in $\mathscr{R}^2$?

Spanning Set: Vectors $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k$ span $\mathscr{R}^m$ is any vector in $\mathscr{R}^m$ can be represented as a linear combination of $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k$, i.e., $\sum_{i=1}^{m} \lambda_i \mathbf{a_i}$ can represent any vector in $\mathscr{R}^m$.

Basis: Vectors $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k$ form a basis of $\mathscr{R}^m$ if they span $\mathscr{R}^m$ and any smaller subset of these vectors does not span $\mathscr{R}^m$. Vectors $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k$ can only form a basis of $\mathscr{R}^m$ if $k = m$ and they are linearly independent.

Convex and Polyhedral Sets

Convex Set: Set $\mathscr{S}$ in $\mathscr{R}^n$ is a *convex set* if a line segment joining any pair of points $\mathbf{a}_1$ and $\mathbf{a}_2$ in $\mathscr{S}$ is completely contained in $\int$, that is, $\lambda\mathbf{a}_1 + (1-\lambda)\mathbf{a}_2 \in \mathscr{S}, \forall\lambda \in [0,1]$.

Hyperplanes and Half-Spaces: A hyperplane in $\mathscr{R}^n$ divides $\mathscr{R}^n$ into 2 half-spaces (like a line does in $\mathscr{R}^2$). A hyperplane is the set $\{\mathbf{x} : \mathbf{px} = k\}$, where $\mathbf{p}$ is the gradient to the hyperplane (i.e., the coefficients of our linear expression). The corresponding half-spaces is the set of points $\{\mathbf{x} : \mathbf{px} \geq k\}$ and $\{\mathbf{x} : \mathbf{px} \leq k\}$.

Polyhedral Set: A *polyhedral set* (or polyhedron) is the set of points in the intersection of a finite set of half-spaces. Set $\mathscr{S} = \{\mathbf{x} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq 0\}$, where $\mathbf{A}$ is an $m \times n$ matrix, $\mathbf{x}$ is an n-vector, and $\mathbf{b}$ is an m-vector, is a *polyhedral set* defined by $m+n$ hyperplanes (i.e., the intersection of $m+n$ half-spaces).

- Polyhedral sets are convex.
- A polytope is a bounded polyhedral set.
- A polyhedral cone is a polyhedral set where the hyperplanes (that define the half-spaces) pass through the origin, thus $\mathscr{C} = \{\mathbf{x} : \mathbf{Ax} \leq 0\}$ is a polyhedral cone.

Edges and Faces: An *edge* of a polyhedral set $\mathscr{S}$ is defined by $n-1$ hyperplanes, and a *face* of $\mathscr{S}$ by one of more defining hyperplanes of $\mathscr{S}$, thus an extreme point and an edge are faces (an extreme point is a zero-dimensional face and an edge a one-dimensional face). In $\mathscr{R}^2$ faces are only edges and extreme points, but in $\mathscr{R}^3$ there is a third type of face, and so on...

Extreme Points: $\mathbf{x} \in \mathscr{S}$ is an extreme point of $\mathscr{S}$ if:

Definition 1: $\mathbf{x}$ is not a convex combination of two other points in $\mathscr{S}$, that is, all line segments that are completely in $\mathscr{S}$ that contain $\mathbf{x}$ must have $\mathbf{x}$ as an endpoint.

Definition 2: $\mathbf{x}$ lies on n linearly independent defining hyperplanes of $\mathscr{S}$.

If more than n hyperplanes pass through an extreme points then it is a degenerate extreme point, and the polyhedral set is considered degenerate. This just adds a bit of complexity to the algorithms we will study, but it is quite common.

Unbounded Sets:

Rays: A ray in $\mathscr{R}^n$ is the set of points $\{\mathbf{x} : \mathbf{x}_0 + \lambda\mathbf{d},\ \lambda \geq 0\}$, where $\mathbf{x}_0$ is the vertex and $\mathbf{d}$ is the direction of the ray.

Convex Cone: A *Convex Cone* is a convex set that consists of rays emanating from the origin. A convex cone is completely specified by its extreme directions. If $\mathscr{C}$ is convex cone, then for any $\mathbf{x} \in \mathscr{C}$ we have

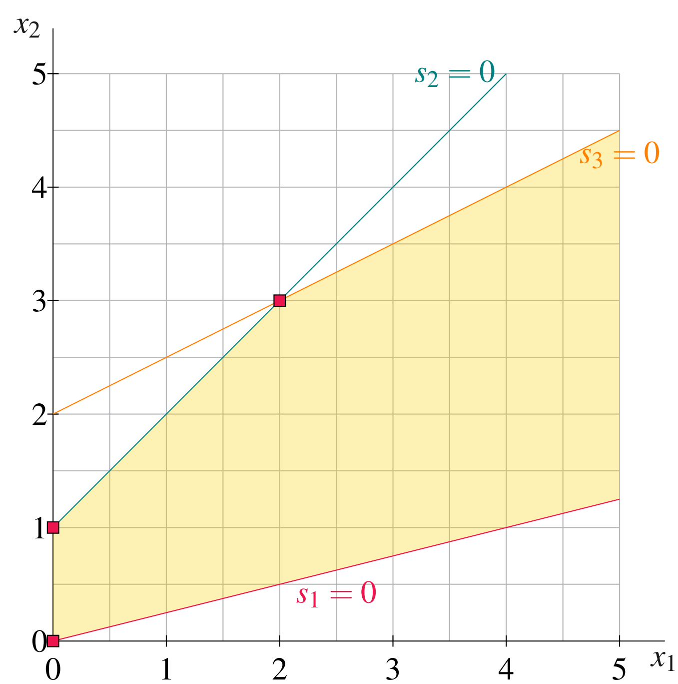
$\lambda \mathbf{x} \in \mathscr{C}$, $\lambda \geq 0$.

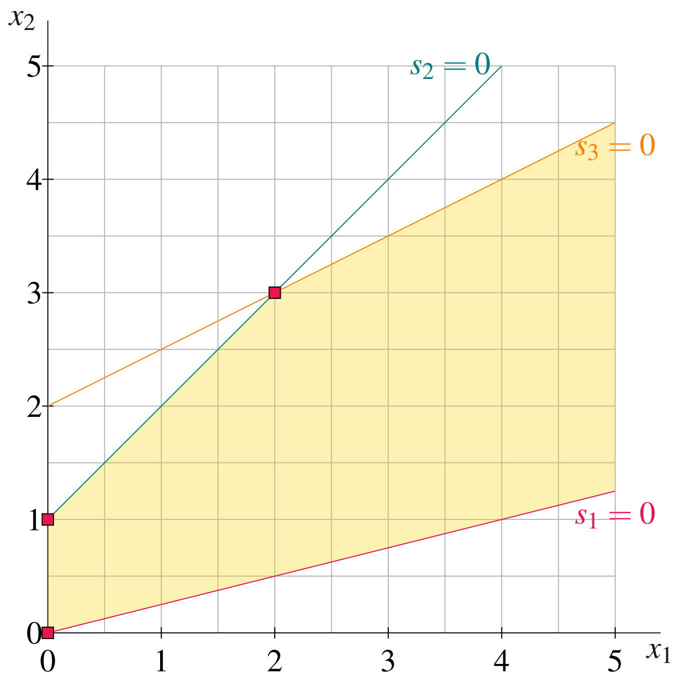
Unbounded Polyhedral Sets: If $\mathscr{S}$ is unbounded, it will have *directions*. $\mathbf{d}$ is a direction of $\mathscr{S}$ only if $\mathbf{A}\mathbf{x}+\lambda\mathbf{d} \leq \mathbf{b}, \mathbf{x}+\lambda\mathbf{d} \geq 0$ for all $\lambda \geq 0$ and all $\mathbf{x} \in \mathscr{S}$. In other words, consider the ray $\{\mathbf{x} : \mathbf{x}_0 + \lambda\mathbf{d},\ \lambda \geq 0\}$ in $\mathscr{R}^n$, where $\mathbf{x}_0$ is the vertex and $\mathbf{d}$ is the direction of the ray. $\mathbf{d} \neq 0$ is a **direction** of set $\mathscr{S}$ if for each $\mathbf{x}_0$ in $\mathscr{S}$ the ray $\{\mathbf{x}_0 + \lambda\mathbf{d},\ \lambda \geq 0\}$ also belongs to $\mathscr{S}$.

Extreme Directions: An *extreme direction* of $\mathscr{S}$ is a direction that *cannot* be represented as positive linear combination of other directions of $\mathscr{S}$. A non-negative linear combination of extreme directions can be used to represent all other directions of $\mathscr{S}$. A polyhedral cone is completely specified by its extreme directions.

Let's define a procedure for finding the extreme directions, using the following LP's feasible region. Graphically, we can see that the extreme directions should follow the the $s_1 = 0$ (red) line and the $s_3 = 0$ (orange) line.

$$
\begin{aligned}
\max\ \ & z = -5x_1 - x_2 \\
\text{s.t.}\ \ & x_1 - 4x_2 + s_1 = 0 \\
& -x_1 + x_2 + s_2 = 1 \\
& -x_1 + 2x_2 + s_3 = 4 \\
& x_1, x_2, s_1, s_2, s_3 \geq 0.
\end{aligned}
$$



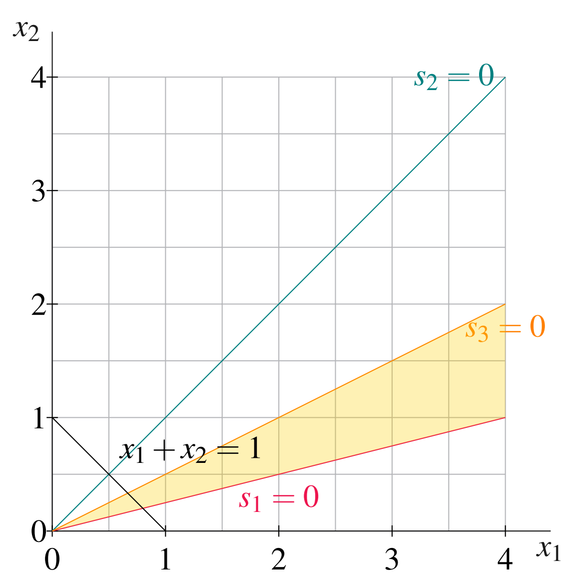


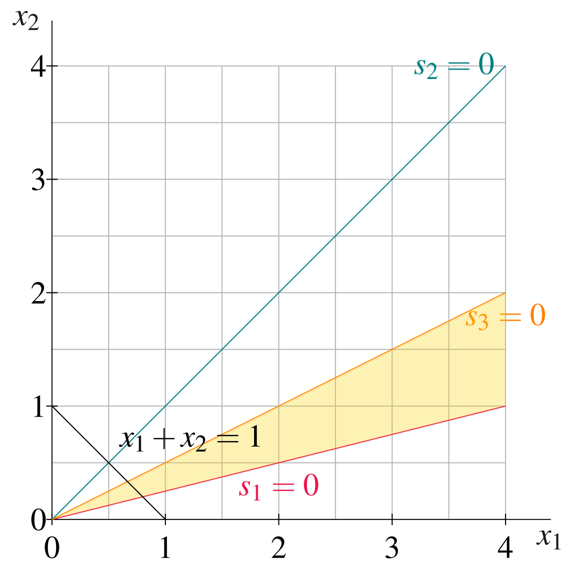
E.g., consider the $s_3 = 0$ (orange) line, to find the extreme direction start at extreme point (2,3) and find another feasible point on the orange line, say (4,4) and subtract (2,3) from (4,4), which yields (2,1).

This is related to the slope in two-dimensions, as discussed in class, the rise is 1 and the run is 2. So this direction has a slope of 1/2, but this does not carry over easily to higher dimensions where directions cannot be defined by a single number.

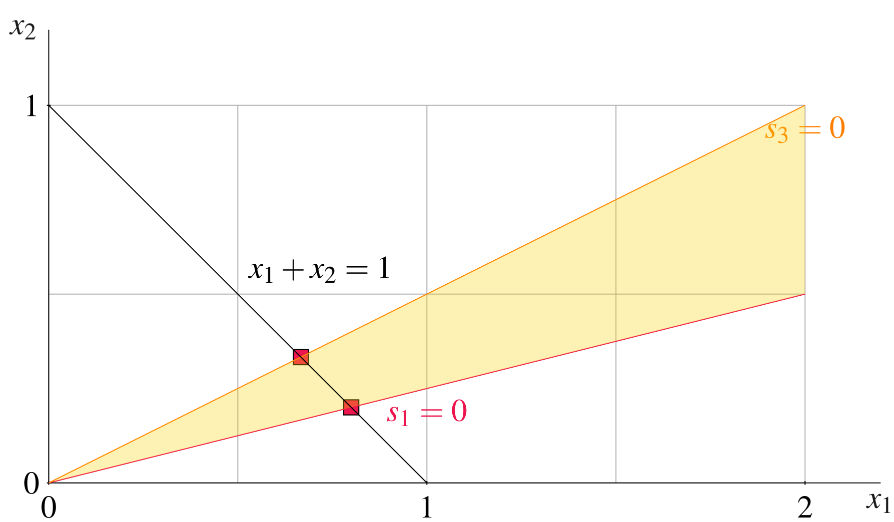
To find the extreme directions we can change the right-hand-side to $\mathbf{b} = 0$, which forms a polyhedral cone (in yellow), and then add the constraint $x_1 + x_2 = 1$. The intersection of the cone and $x_1 + x_2 = 1$ form a line segment.

$$
\begin{aligned}
\max\ & z = -5x_1 - x_2 \\
\text{s.t. } & x_1 - 4x_2 + s_1 = 0 \\
& -x_1 + x_2 + s_2 = 0 \\
& -x_1 + 2x_2 + s_3 = 0 \\
& x_1 + x_2 = 1 \\
& x_1, x_2, s_1, s_2, s_3 \geq 0.
\end{aligned}
$$





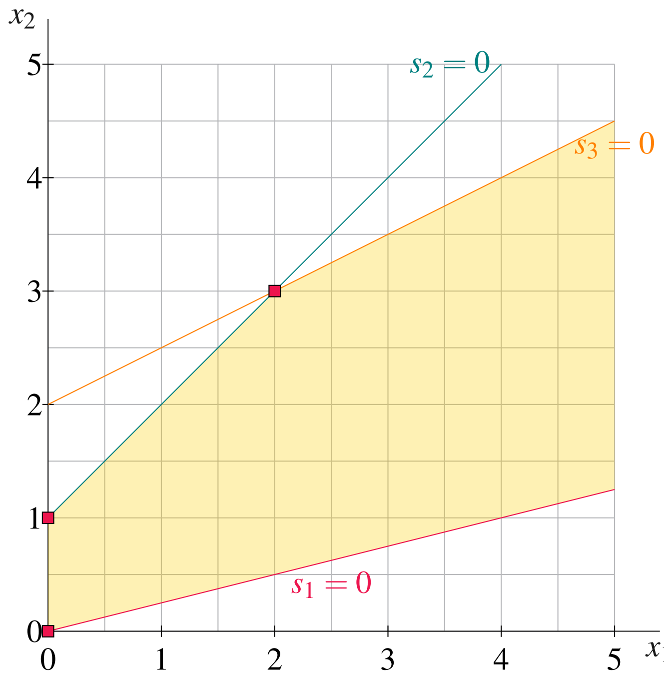
Magnifying for clarity, and removing the $s_2 = 0$ (teal) line, as it is redundant, and marking the extreme points of the new feasible region, (4/5, 1/5) and (2/3, 1/3), with red boxes, we have:



The extreme directions are thus (4/5, 1/5) and (2/3, 1/3).

Representation Theorem: Let $\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x_k}$ be the set of extreme points of $\mathscr{S}$, and if $\mathscr{S}$ is unbounded, $\mathbf{d}_1, \mathbf{d}_2, \cdots \mathbf{d_l}$ be the set of extreme directions. Then any $\mathbf{x} \in \mathscr{S}$ is equal to a convex combination of the extreme points and a non-negative linear combination of the extreme directions: $\mathbf{x} = \sum_{j=1}^{k} \lambda_j \mathbf{x_j} + \sum_{j=1}^{l} \mu_j \mathbf{d_j}$, where $\sum_{j=1}^{k} \lambda_j = 1$, $\lambda_j \geq 0$, $\forall j = 1, 2, \cdots, k$, and $\mu_j \geq 0$, $\forall j = 1, 2, \cdots, l$.

$$
\begin{aligned}
\max \quad & z = -5x_1 - x_2 \\
\text{s.t.} \quad & x_1 - 4x_2 + s_1 = 0 \\
& -x_1 + x_2 + s_2 = 1 \\
& -x_1 + 2x_2 + s_3 = 4 \\
& x_1, x_2, s_1, s_2, s_3 \geq 0.
\end{aligned}
$$

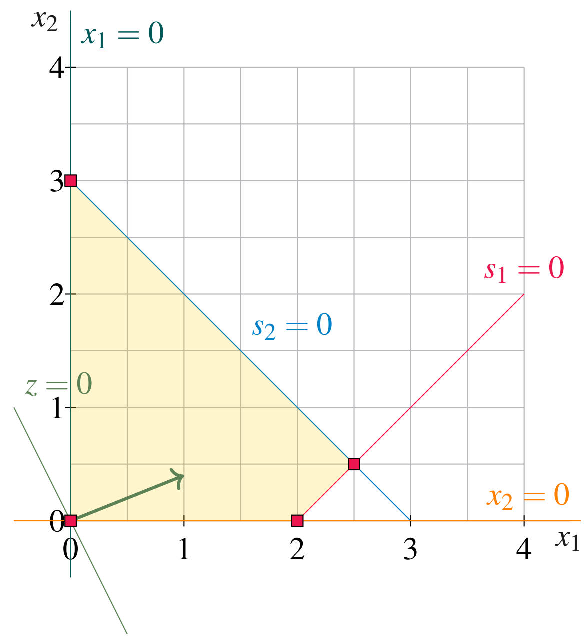


Represent point (1/2, 1) as a convex combination of the extreme points of the above LP. Find λs to solve the following system of equations:

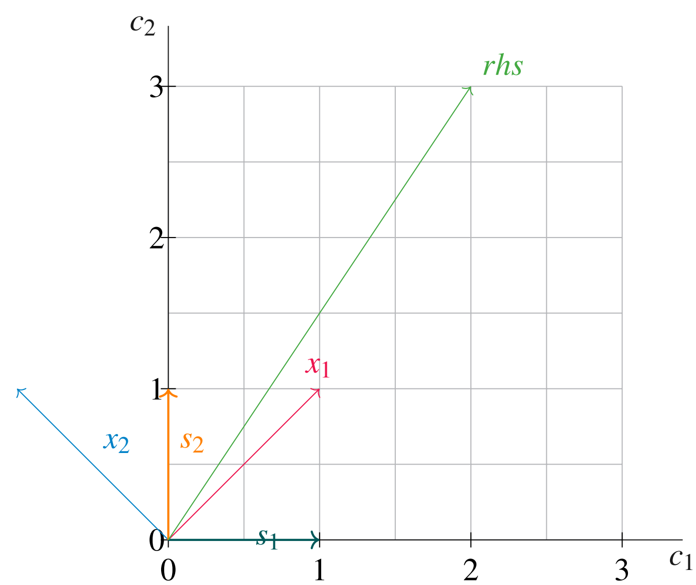
$$
\lambda_1 \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \lambda_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \lambda_3 \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 1 \end{bmatrix}
$$

The Variable (Canonical Form) and Requirement Space

$$\begin{aligned} \max \quad & z = 2x_1 + x_2 \\ \text{s.t.} \quad & x_1 - x_2 + s_1 = 2 \\ & x_1 + x_2 + s_2 = 3 \\ & x_1, x_2, s_1, s_2 \geq 0. \end{aligned}$$



$$\begin{aligned} \max \quad & z = 2x_1 + x_2 \\ \text{s.t.} \quad & x_1 - x_2 + s_1 = 2 \\ & x_1 + x_2 + s_2 = 3 \\ & x_1, x_2, s_1, s_2 \geq 0. \end{aligned}$$



Tableaus

After putting an LP into standard form, we can put the system of equations into a table form, the “tableau".

$$\begin{aligned} \max \quad & z = 2x_1 + x_2 \\ \text{s.t.} \quad & x_1 - x_2 + s_1 = 2 \\ & x_1 + x_2 + s_2 = 3 \\ & x_1, x_2, s_1, s_2 \geq 0. \end{aligned}$$

max	z	x_1	x_2	x_3	x_4	rhs
(r0 - z)	1	-2	-1	0	0	0
(r1 - x_3)	0	1	-1	1	0	2
(r2 - x_4)	0	1	1	0	1	3

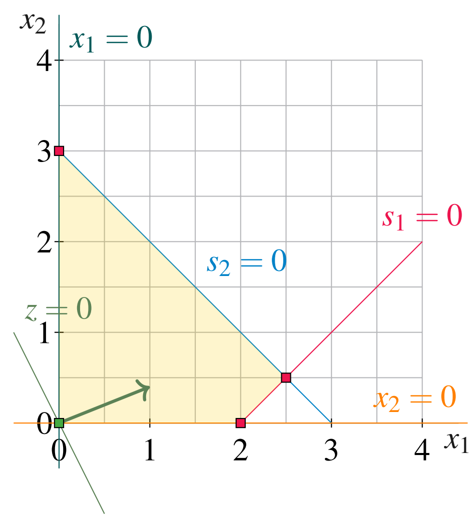
Why are the coefficients negative in row zero, we change $z = 2x_1 + x_2$ to $z - 2x_1 - x_2 = 0$ so we have only constants on the right-hand-side (rhs).

This tableau represents a basic solution, because it contains an identity matrix. The basic variables are those variables having columns in the identity matrix (here, x_3 and x_4), and it is feasible because the rhs for row $1-m$ are non-negative.

We can consider z a permanent member of an expanded basis if we treat row zero like any other row (although the rhs of row 0 can be negative).

Basic Solutions and Extreme Points

max	z	x_1	x_2	x_3	x_4	rhs
(r0 - z)	1	-2	-1	0	0	0
(r1 - x_3)	0	1	-1	1	0	2
(r2 - x_4)	0	1	1	0	1	3



Here the basic variables are $x_3 = 2$ and $x_4 = 3$, and the z-value of objective function value is 0.

Let's go to the extreme point (2,0) which has basic variables x_1 and x_4 (this new extreme point is adjacent to the extreme point (0,0). Why?

max	z	x_1	x_2	x_3	x_4	rhs
(r0 - z)	1	-2	-1	0	0	0
(r1 - x_3)	0	1	-1	1	0	2
(r2 - x_4)	0	1	1	0	1	3

First get a 1 coefficent in row 1 in the x_1 column by multiplying row 1 by a scalar (no action needed, already equals one).

max	z	x_1	x_2	x_3	x_4	rhs
(r0 - z)	1	-2	-1	0	0	0
(r1 - x_3)	0	1	-1	1	0	2
(r2 - x_4)	0	1	1	0	1	3

Then use row 1 to zero out the row 0 coefficient for x_1 by multiplying row 1 by 2 and adding it to row 0 to get a new row 0.

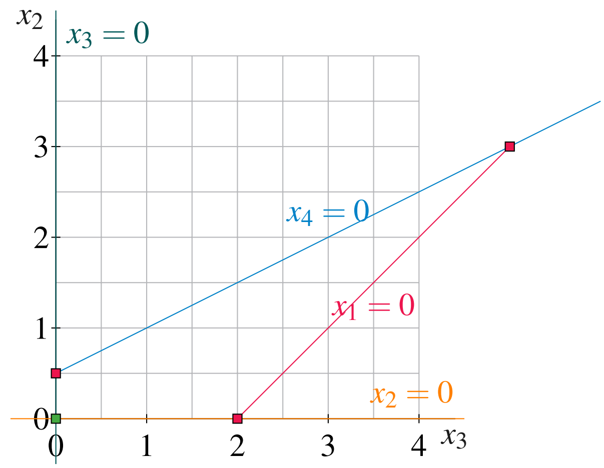
max	z	x_1	x_2	x_3	x_4	rhs
(r0 -z)	1	0	-3	2	0	4
(r1 - x_3)	0	1	-1	1	0	2
(r2 - x_4)	0	1	1	0	1	3

Lastly use row 1 to zero out the row 20 coefficient for x_1 by multiplying row 1 by -1 and adding it to row 2 to get a new row 2.

max	z	x_1	x_2	x_3	x_4	rhs
(r0 -z)	1	0	-3	2	0	4
(r1 - x_3)	0	1	-1	1	0	2
(r2 - x_4)	0	0	2	-1	1	1

The nonbasic variables for this tableau are x_2 and x_3, so we can graph the new LP in the nonbasic variable space.

max	z	x_1	x_2	x_3	x_4	rhs
(r0 -z)	1	0	-3	2	0	4
(r1 - x_3)	0	1	-1	1	0	2
(r2 - x_4)	0	0	2	-1	1	1



<u>Matrix Math</u>

An $m \times n$ matrix is an array of real numbers with m rows and n columns. Any matrix can be represented by its constituent set of row or column vectors.

$$\mathbf{A} = \begin{bmatrix} 1 & 3 \\ 6 & 4 \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{a}^1 \\ \mathbf{a}^2 \end{bmatrix},$$

where $\mathbf{a}_1 = \begin{bmatrix} 1 \\ 6 \end{bmatrix}$, $\mathbf{a}_2 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$, $\mathbf{a}^1 = \begin{bmatrix} 1 & 3 \end{bmatrix}$, and $\mathbf{a}^2 = \begin{bmatrix} 6 & 4 \end{bmatrix}$. Additionally, $a_{11} = 1$, $a_{12} = 3$, $a_{21} = 6$, $a_{22} = 4$.

Matrix Addition: Two matrices of the same dimension can be added componentwise, thus $\mathbf{C} = \mathbf{A} + \mathbf{B}$ means that $c_{ij} = a_{ij} + b_{ij}$.

Scalar Multiplication: Just like it sounds. If k is a scalar, then $k\mathbf{A}$ means that every component of $\mathbf{A}$ is multiplied by k.

Matrix Multiplication: $\mathbf{A}$ is a $m \times n$ matrix and $\mathbf{B}$ is a $p \times q$ matrix. $\mathbf{AB}$ (matrix multiplication) is only defined if $n = p$ and the result is a $m \times q$ matrix, $\mathbf{BA}$ is only defined if $q = m$ and the result is a $p \times n$. $\mathbf{AB}$ is not necessarily equal to $\mathbf{BA}$, thus $\mathbf{C} = \mathbf{AB}$ where $c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$, $i = 1, \cdots, m$, $j = 1, \cdots, p$, e.g., c_{11} is the sum of the the componentwise multiplication of the first row of $\mathbf{A}$ and the first column of $\mathbf{B}$.

Identity Matrix: A square matrix (denoted by $\mathbf{I}$) with all zero components, except for the diagonal:
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Elementary Matrix Operations: These operations are used to solve systems of linear equations or inverting a matrix. The three operations are as follows (for any matrix $\mathbf{A}$):

- Interchange two rows of $\mathbf{A}$.
- Multiply a row by a nonzero scalar.
- Replace row i with row i plus row j multiplied by a nonzero scalar.

Inverting a Matrix:

$$\mathbf{A} = \begin{bmatrix} 3 & 9 & 2 \\ 1 & 1 & 1 \\ 5 & 4 & 7 \end{bmatrix} \textit{ and } \mathbf{A}^{-1} = \begin{bmatrix} -\frac{3}{11} & 5 & -\frac{7}{11} \\ \frac{2}{11} & -1 & \frac{1}{11} \\ \frac{1}{11} & -3 & \frac{6}{11} \end{bmatrix}$$

To find $\mathbf{A}^{-1}$ using elementary row operations to transform $\mathbf{A}$ into an identity matrix, while performing these same operations on the attached identity matrix.

$$\left[\begin{array}{ccc|ccc} 3 & 9 & 2 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 5 & 4 & 7 & 0 & 0 & 1 \end{array}\right] \Rightarrow \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & -\frac{3}{11} & 5 & -\frac{7}{11} \\ 0 & 1 & 0 & \frac{2}{11} & -1 & \frac{1}{11} \\ 0 & 0 & 1 & \frac{1}{11} & -3 & \frac{6}{11} \end{array}\right].$$

Rank of a Matrix: $\mathbf{A}$ is a $m \times n$ matrix then $\text{rank}(\mathbf{A}) \leq min\{m,\ n\}$, if $\text{rank}(\mathbf{A}) = min\{m,\ n\}$, then $\mathbf{A}$ is of full rank. If $\mathbf{A}$ is not full rank, but of rank k, where $k < min\{m,\ n\}$, then using the elementary row operations, we can transform $\mathbf{A}$ to the following: $\begin{bmatrix} \mathbf{I_k} & \mathbf{Q} \\ 0 & 0 \end{bmatrix}$

4.2.3. Linear Optimization Theory

Consider an arbitrary LP, which we will call the primal (P):

$$(P) : \max\{\mathbf{cx} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq 0\},$$

where $\mathbf{A}$ is an $m \times n$ matrix, and $\mathbf{x}$ is a n element column vector. Every prmal LP has a related LP, which we call the dual, the dual of (P) is:

$$(D) : \min\{\mathbf{wb} : \mathbf{wA} \geq \mathbf{c}, \mathbf{w} \geq 0\}.$$

Before we discuss properties of duality, and why it is important, we start with how to formulate the dual for any given LP. If the LP has a different form like P, we find the dual based on the P and D example above. If the LP does not have this form, we can transform it to this form, or use the rules in the following table, first noting that:

- The dual of problem D is problem P.
- Each primal constraint has an associated dual variable (w_i) and each dual constraint has an associated primal variable (x_i).
- When the primal is a maximization, the dual is a minimization, and vice versa.

$\max \mathbf{cx}$:	$\min \mathbf{wb}$:
$\mathbf{a}_{1*}\mathbf{x} \leq b_1 \ (w_1 \geq 0)$	$\mathbf{wa}_{*1} \geq c_1 \ (x_1 \geq 0)$
$\mathbf{a}_{2*}\mathbf{x} = b_2 \ (w_2\ urs)$	$\mathbf{wa}_{*2} = c_2 \ (x_2\ urs)$
$\mathbf{a}_{3*}\mathbf{x} \geq b_3 \ (w_3 \leq 0)$	$\mathbf{wa}_{*3} \leq c_3 \ (x_3 \leq 0)$
$\vdots$	$\vdots$
$x_1 \geq 0, x_2\ urs, x_3 \leq 0, \cdots$	$w_1 \geq 0, w_2\ urs, w_3 \leq 0, \cdots$

To illustrate the relationship between the primal and dual, consider this production problem we previously formulated:

Production Problem: You have 21 units of transparent aluminum alloy (TAA), LazWeld1, a joining robot leased for 23 hours, and CrumCut1, a cutting robot leased for 17 hours of aluminum cutting. You also have production code for a bookcase, desk, and cabinet, along with commitments to buy any of these you can produce for \$18, \$16, and \$10 apiece, respectively. A bookcase requires 2 units of TAA, 3 hours of joining, and 1 hour of cutting, a desk requires 2 units of TAA, 2 hours of joining, and 2 hour of cutting, and a cabinet requires 1 unit of TAA, 2 hours of joining, and 1 hour of cutting. Formulate an LP to maximize your revenue given your current resources.

<u>Decision variables:</u>
x_i : number of units of product i to produce,
$\forall i = \{bookcase,\ desk,\ cabinet\}$.

$$\begin{aligned}
\max z = 18x_1 + 16x_2 + 10x_3 :& \\
2x_1 + 2x_2 + 1x_3 \leq 21 && (TAA) \\
3x_1 + 2x_2 + 2x_3 \leq 23 && (LazWeld1) \\
1x_1 + 2x_2 + 1x_3 \leq 17 && (CrumCut1) \\
x_1, x_2, x_3 \geq 0.
\end{aligned}$$

Considering the formulation above as the primal, consider a new, related, problem: You have an offer to buy all your resources (the leased hours for the two robots, and the TAA). Formulate an LP to find the minimum value of the resources given the above plans for the three products and commitments to buy them.

<u>Decision variables:</u>
w_i : selling price, per unit, for resource i, $\forall i = \{TAA,\ LazWeld1,\ CrumCut1\}$.

$$\begin{aligned}
\min\ \ 21w_1 + 23w_2 + 17w_3 :& \\
2w_1 + 3w_2 + 1w_3 \geq 18& \\
2w_1 + 2w_2 + 2w_3 \geq 16& \\
1w_1 + 2w_2 + 1w_3 \geq 10& \\
w_1, w_2, w_3 \geq 0.&
\end{aligned}$$

Define $\mathbf{w} = \mathbf{c}_B\mathbf{B}^{-1}$ as the vector of *shadow prices*, where w_i represents the change in the objective function value caused by a unit change to the associated b_i parameter (i.e., increasing the amount of resource i by one unit, see dual objective function).

Consider the following primal tableau (where z_p is the primal objective function value) for $(P) : (max\ \{\mathbf{cx} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq 0\})$

	z_P	x_i	rhs
z_P	1	$\mathbf{c_B B^{-1} a_i} - c_i$	$\mathbf{c_B B^{-1} b}$
BV	0	$\mathbf{B^{-1} a_i}$	$\mathbf{B^{-1} b}$

Observe that if a basis for P is optimal, then the row zero coefficients for the variables are greater than, or equal to, zero, that is, $c_B B^{-1} a_i - c_i \geq 0$ for each x_i (if the variable is a slack, this simplifies to $c_B B^{-1} \geq 0$).

Substituting $w = c_B B^{-1}$ we get $\mathbf{wA} \geq \mathbf{c}, \mathbf{w} \geq 0$ which corresponds to dual feasibility.

$$(D) : \min\{\mathbf{wb} : \mathbf{wA} \geq \mathbf{c}, \mathbf{w} \geq 0\}.$$

Weak Duality Property
If $\mathbf{x}$ and $\mathbf{w}$ are feasible solutions to P and D, respectively, then $\mathbf{cx} \leq \mathbf{wAx} \leq \mathbf{wb}$.

$$(P) : \max\{\mathbf{cx} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq 0\}.$$
$$(D) : \min\{\mathbf{wb} : \mathbf{wA} \geq \mathbf{c}, \mathbf{w} \geq 0\}.$$

This implies that the objective function value for a feasible solution to P is a lower bound on the objective function value for the optimal solution to D, and the objective function value for a feasible solution to D is an upper bound on the objective function value for the optimal solution to P.

Thus if the objective function values are equal, i.e., $\mathbf{cx} = \mathbf{wb}$, then the solutions $\mathbf{x}$ and $\mathbf{w}$ are optimal.

Fundamental Theorem of Duality
For problems P and D (i.e., any primal dual set) exactly one of the following is true:

1. Both have optimal solutions $\mathbf{x}$ and $\mathbf{w}$ where $\mathbf{cx} = \mathbf{wb}$.
2. One problem is unbounded (i.e., the objective function value can become arbitrarily large for a maximization, or arbitrarily small for a minimization), and the other is infeasible.
3. Both are infeasible.

4.2.3.1. Optimality Conditions

Farka's Lemma
Consider the following two systems:

1. $\mathbf{Ax} \geq 0$, $\mathbf{cx} < 0$.
2. $\mathbf{wA} = \mathbf{c}$, $\mathbf{w} \geq 0$.

Farka's Lemma - exactly one of these systems has a solution.

Suppose system 1 has x as a solution:

- If $\mathbf{w}$ were a solution to system 2, then post-multiplying each side of $\mathbf{wA} = \mathbf{c}$ by $\mathbf{x}$ would yield $\mathbf{wAx} = \mathbf{cx}$.
- Since $\mathbf{Ax} \geq 0$ and $\mathbf{w} \geq 0$, this implies that $\mathbf{cx} \geq 0$, which violates $\mathbf{cx} < 0$.
- Thus we show that if system 1 has a solution, system 2 cannot have one.

Suppose system 1 has no solution:

- Consider the following LP: $\min\{\mathbf{cx} : \mathbf{Ax} \geq 0\}$.
- The optimal solution is $\mathbf{cx} = 0$ and $\mathbf{x} = 0$.
- The LP in standard form (substitute $\mathbf{x} = \mathbf{x}' - \mathbf{x}''$, $\mathbf{x}' \geq 0$ and $\mathbf{x}'' \geq 0$ and add $\mathbf{x^s} \geq 0$) follows:
$$\min\{\mathbf{cx}' - \mathbf{cx}'' : \mathbf{Ax}' - \mathbf{Ax}'' - \mathbf{x^s} = 0, \mathbf{x}', \mathbf{x}'', \mathbf{x^s} \geq 0\}$$
- $\mathbf{x}' = 0$, $\mathbf{x}'' = 0$, $\mathbf{x^s} = 0$ is an optimal extreme point solution.
- Using $\mathbf{x^s}$ as an initial feasible basis, solve with the simplex algorithm (with cycling prevention) to find a basis where $\mathbf{c_B B}^{-1}\mathbf{a_i} - c_i \leq 0$ for all variables. Define $\mathbf{w} = \mathbf{c_B B}^{-1}$.
- This yields $\mathbf{wA} - \mathbf{c} \leq 0$, $-\mathbf{wA} + \mathbf{c} \leq 0$, $-\mathbf{w} \leq 0\}$, from the columns for variables $\mathbf{x}'$, $\mathbf{x}''$, $\mathbf{x^s}$, respectively. Thus, $\mathbf{w} \geq 0$ and $\mathbf{wA} = \mathbf{c}$, and system 2 has a solution.

Karush-Kuhn-Tucker (KKT) Conditions

$$(P) : \max\{\mathbf{cx} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq 0\}.$$
$$(D) : \min\{\mathbf{wb} : \mathbf{wA} \geq \mathbf{c}, \mathbf{w} \geq 0\}.$$

For problems P and D, with solutions $\mathbf{x}$ and $\mathbf{w}$, respectively, we have the following conditions, which for LPs are necessary and sufficient conditions for optimality:

1. $\mathbf{Ax} \leq \mathbf{b}$, $\mathbf{x} \geq 0$ (primal feasibility).

2. $\mathbf{wA} \geq \mathbf{c}$, $\mathbf{w} \geq 0$ (dual feasibility).
3. $\mathbf{w}(\mathbf{Ax} - \mathbf{b}) = 0$ and $\mathbf{x}(\mathbf{c} - \mathbf{wA}) = 0$ (complementary slackness).

Note we can rewrite the third condition as $\mathbf{w}(\mathbf{Ax} - \mathbf{b}) = \mathbf{wx^s} = 0$ and $\mathbf{x}(c - \mathbf{wA}) = \mathbf{xw^s} = 0$, where $\mathbf{x^s}$ and $\mathbf{w^s}$ are the slack variables for the primal and dual problems, respectively.

Why do the KKT conditions hold?

Suppose that the LP $\min\{\mathbf{cx} : \mathbf{Ax} \geq \mathbf{b}, \mathbf{x} \geq 0\}$ has an optimal solution $\mathbf{x}^*$ (the dual is $\max\{\mathbf{wb} : \mathbf{wA} \leq \mathbf{c}, \mathbf{w} \geq 0\}$).

- Since $\mathbf{x}^*$ is optimal there is no direction $\mathbf{d}$ such that $\mathbf{c}(\mathbf{x}^* + \lambda \mathbf{d}) < \mathbf{cx}^*$, $\mathbf{A}(\mathbf{x}^* + \lambda \mathbf{d}) \geq \mathbf{b}$, and $\mathbf{x}^* + \lambda \mathbf{d} \geq 0$ for $\lambda > 0$.
- Let $\mathbf{Gx} \geq \mathbf{g}$ be the binding inequalities in $\mathbf{Ax} \geq \mathbf{b}$ and $\mathbf{x} \geq 0$ for solution $\mathbf{x}^*$ that is, $\mathbf{Gx}^* = \mathbf{g}$.
- Based on the optimality of $\mathbf{x}^*$, there is no direction $\mathbf{d}$ at $\mathbf{x}^*$ such that $\mathbf{cd} < 0$ and $\mathbf{Gd} \geq 0$ (else we could improve the solution).
- Based on Farka's Lemma, if the system $\mathbf{cd} < 0$, $\mathbf{Gd} \geq 0$ does not have a solution, the system $\mathbf{wG} = \mathbf{c}$, $\mathbf{w} \geq 0$ must have a solution.
- $\mathbf{G}$ is composed of rows from $\mathbf{A}$ where $\mathbf{a_{i*}x}^* = b_i$ and vectors $\mathbf{e_i}$ for any $x_i^* = 0$.
- We can divide the $\mathbf{w}$ into two sets:
  - $\{w_i,\ i : \mathbf{a_{i*}x}^* = b_i\}$ - those corresponding to the binding functional constraints in the primal.
  - $\{w_i^s,\ j : x_i^* = 0\}$ - those corresponding to the binding non-negativity constraints in the primal.
- Thus $\mathbf{G}$ has the columns $\mathbf{a_{i*}^T}$ for w_i and e_i^T for w_i^s.
- Since $\mathbf{wG} = \mathbf{c}$, $\mathbf{w} \geq 0$ must have a solution, this solution is feasible for $\mathbf{wA} \leq \mathbf{c}, \mathbf{w} \geq 0$ where w_i^s are added slacks. Thus, $\mathbf{G}$ is missing some columns from $\mathbf{A}$ (and thus some w variables) and some slack variables if $\mathbf{wA} \leq \mathbf{c}, \mathbf{w} \geq 0$ were put into standard form, but those are not needed for feasibility based on the result, and thus can be thought of as set to zero, giving us complementary slackness.

Example: Consider a production LP (the primal P) where the variables represent the amount of three products to produce, using three resources, represented by the functional constraints. In standard form P and D have x_4^s, x_5^s, x_6^s and w_4^s, w_5^s, w_6^s as slack variables, respectively.

Decision variables:
x_i : number of units of product i to produce, $\forall i = \{1,\ 2,\ 3\}$.

$$\begin{aligned}(P): \max\ & z_P = 18x_1 + 16x_2 + 10x_3 \\ s.t.\ & 2x_1 + 2x_2 + 1x_3 + x_4^s = 21 \ \ (w_1) \\ & 3x_1 + 2x_2 + 2x_3 + x_5^s = 23 \ \ (w_2) \\ & 1x_1 + 2x_2 + 1x_3 + x_6^s = 17 \ \ (w_3) \\ & x_1, x_2, x_3, x_4^s, x_5^s, x_6^s \geq 0.\end{aligned}$$

$$\begin{aligned}(D): \min\ & z_D = 21w_1 + 23w_2 + 17w_3 \\ s.t.\ & 2w_1 + 3w_2 + 1w_3 \geq 18 \ \ (x_1) \\ & 2w_1 + 2w_2 + 2w_3 \geq 16 \ \ (x_2) \\ & 1w_1 + 2w_2 + 1w_3 \geq 10 \ \ (x_3) \\ & 1w_1 \geq 0 \\ & 1w_2 \geq 0 \\ & 1w_3 \geq 0 \\ & w_1, w_2, w_3\ urs.\end{aligned}$$

Decision variables:
w_i : unit selling price for resource i, $\forall i = \{1,\ 2,\ 3\}$.

$$\begin{aligned}(D): \min\ & z_D = 21w_1 + 23w_2 + 17w_3 : \\ & 2w_1 + 3w_2 + 1w_3 - w_4^s = 18 \ \ (x_1) \\ & 2w_1 + 2w_2 + 2w_3 - w_5^s = 16 \ \ (x_2) \\ & 1w_1 + 2w_2 + 1w_3 - w_6^s = 10 \ \ (x_3) \\ & w_1, w_2, w_3, w_4^s, w_5^s, w_6^s \geq 0.\end{aligned}$$

The initial basic feasible tableau for the primal, i.e., having the slack variables form the basis, follows:

P : max	z_P	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	rhs
z_P	1	-18	-16	-10	0	0	0	0
x_4^s	0	2	2	1	1	0	0	21
x_5^s	0	3	2	2	0	1	0	23
x_6^s	0	1	2	1	0	0	1	17

$x_1, x_2, x_3 = 0,\ x_4^s = 21,\ x_5^s = 23,\ x_6^s = 17\ z_P = 0$

The following dual tableau **conforms with the primal tableau through complementary slackness**.

$D:\min$	z_D	w_1	w_2	w_3	w_4^s	w_5^s	w_6^s	rhs
z_D	1	-21	-23	-17	0	0	0	0
w_4^s	0	-2	-3	-1	1	0	0	-18
w_5^s	0	-2	-2	-2	0	1	0	-16
w_6^s	0	-1	-2	-1	0	0	1	-10

$w_1, w_2, w_3 = 0$, $w_4^s = -18$, $w_5^s = -16$, $w_6^s = -10$ $z_D = 0$

Complementary slackness: $w_1 x_4^s = 0$, $w_2 x_5^s = 0$, $w_3 x_6^s = 0$, $x_1 w_4^s = 0$, $x_2 w_5^s = 0$, $x_3 w_6^s = 0$.

- If a primal variable is basic, then its corresponding dual variable must be nonbasic, and vise versa.
- The primal is suboptimal, and the dual tableau has a basic infeasible solution.
- Row 0 of the primal tableau has dual variable values in the corresponding primal variable columns.

The primal basis is not optimal, so enter x_1 into the basis, and remove x_5^s, which yields:

P: Max	z_P	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	rhs
z_P	1	0	-4	2	0	6	0	138
x_4^s	0	0	2/3	-1/3	1	-2/3	0	17/3
x_1	0	1	2/3	2/3	0	1/3	0	23/3
x_6^s	0	0	4/3	1/3	0	-1/3	1	28/3

D: Min	z_D	w_1	w_2	w_3	w_4^s	w_5^s	w_6^s	rhs
z_D	1	-17/3	0	-28/3	-23/3	0	0	138
w_2	0	2/3	1	1/3	-1/3	0	0	6
w_5^s	0	-2/3	0	-4/3	-2/3	1	0	-4
w_6^s	0	1/3	0	-1/3	-2/3	0	1	2

The primal tableau does not represent an optimal basic solution, and the dual tableau does not represent a feasible basic solution.

Using Dantzig's rule, we enter x_2 into the basis, and using the ratio test we find that x_6^s leaves the basis. This change in basis yields the following tableau:

P: Max	z_P	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	rhs
z_P	1	0	0	3	0	5	3	166
x_4^s	0	0	0	-1/2	1	-1/2	-1/2	1
x_1	0	1	0	1/2	0	1/2	-1/2	3
x_2	0	0	1	1/4	0	-1/4	3/4	7

D: Min	z_D	w_1	w_2	w_3	w_4^s	w_5^s	w_6^s	rhs
z_D	1	-1	0	0	-3	-7	0	166
w_2	0	1/2	1	0	-1/2	1/4	0	5
w_3	0	1/2	0	1	1/2	-3/4	0	3
w_6^s	0	1/2	0	0	-1/2	-1/4	1	3

<u>Decision variables:</u>
x_i : number of units of product i to produce, $\forall i = \{1,\ 2,\ 3\}$.

$$\begin{aligned}(P):\max\ \ & z_P = 18x_1 + 16x_2 + 10x_3 : \\ & 2x_1 + 2x_2 + 1x_3 + x_4^s = 21 \ \ (w_1) \\ & 3x_1 + 2x_2 + 2x_3 + x_5^s = 23 \ \ (w_2) \\ & 1x_1 + 2x_2 + 1x_3 + x_6^s = 17 \ \ (w_3) \\ & x_1, x_2, x_3, x_4^s, x_5^s, x_6^s \geq 0.\end{aligned}$$

The LP $\max\{\mathbf{cx} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq 0\}$ has an optimal solution $\mathbf{x}^*$ (the dual is $\min\{\mathbf{wb} : \mathbf{wA} \geq \mathbf{c}, \mathbf{w} \geq 0\}$).

- Since $\mathbf{x}^*$ is optimal there is no direction $\mathbf{d}$ such that $\mathbf{c}(\mathbf{x}^* + \lambda\mathbf{d}) > \mathbf{cx}^*$, $\mathbf{A}(\mathbf{x}^* + \lambda\mathbf{d}) \leq \mathbf{b}$, and $\mathbf{x}^* + \lambda\mathbf{d} \geq 0$ for $\lambda > 0$.
- Let $\mathbf{Gx} \leq \mathbf{g}$ be the binding inequalities in $\mathbf{Ax} \leq \mathbf{b}$ and $\mathbf{x} \geq 0$ for solution $\mathbf{x}^*$, that is, $\mathbf{Gx}^* = \mathbf{g}$.

  For our example,

$$\mathbf{G}|\mathbf{g} = \left[\begin{array}{ccc|c} 3 & 2 & 2 & 23 \\ 1 & 2 & 1 & 17 \\ 0 & 0 & -1 & 0 \end{array}\right]$$

- Based on the optimality of $\mathbf{x}^*$, there is no direction $\mathbf{d}$ at $\mathbf{x}^*$ such that $\mathbf{cd} > 0$ and $\mathbf{Gd} \leq 0$ (this includes $\mathbf{d} \leq 0$) (else we could improve the solution).
- From Farka's Lemma, if the system $\mathbf{cd} > 0$, $\mathbf{Gd} \leq 0$ does not have a solution, the system $\mathbf{wG} = \mathbf{c}$, $\mathbf{w} \geq 0$ must have a solution.

$$\begin{aligned} & 3w_2 + 1w_3 = 18 \ \ (x_1) \\ & 2w_2 + 2w_3 = 16 \ \ (x_2) \\ & 2w_2 + 1w_3 - w_6^s = 10 \ \ (x_3) \\ & w_2, w_3, w_6^s, \geq 0.\end{aligned}$$

D: Min	z_D	w_1	w_2	w_3	w_4^s	w_5^s	w_6^s	rhs
z_D	1	-1	0	0	-3	-7	0	166
w_2	0	1/2	1	0	-1/2	1/4	0	5
w_3	0	1/2	0	1	1/2	-3/4	0	3
w_6^s	0	1/2	0	0	-1/2	-1/4	1	3

Challenge 1: Solve the following LP (as represented in the tableau), using the given tableau as a starting point. Provide the details of the algorithm to do so, and make it valid for both maximization and minimization problems.

D : min	z_D	w_1	w_2	w_3	w_4^s	w_5^s	w_6^s	rhs
z_D	1	-21	-23	-17	0	0	0	0
w_4^s	0	-2	-3	-1	1	0	0	-18
w_5^s	0	-2	-2	-2	0	1	0	-16
w_6^s	0	-1	-2	-1	0	0	1	-10

Challenge 2: Given the following optimal tableau to our production LP, we can buy 12 units of resource 2 for $4 a unit. Should we, please provide the analysis needed to make this decision.

P : max	z_P	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	rhs
z_P	1	0	0	3	0	5	3	166
x_4^s	0	0	0	-1/2	1	-1/2	-1/2	1
x_1	0	1	0	1/2	0	1/2	-1/2	3
x_2	0	0	1	1/4	0	-1/4	3/4	7

4.2.4. Solution Algorithms

We start with some preliminaries, and then discuss the simplex algorithm, assuming an initial basic feasible solution, including tableau formulas. Next two extensions to the algorithm, for finding an initial basic feasible solution, are discussed. We then explore a useful algorithm for solving certain LPs that have“too many" columns.

Consider an LP $\max\{\mathbf{cx} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq 0\}$ in standard form, where:

- $\mathbf{A}$ is an $m \times n$ matrix of rank m and $n \geq m$; $\mathbf{A}$ consists of n column vectors, $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \cdots, \mathbf{a_n}$.
- $\mathbf{c}$ and $\mathbf{x}$ are n-vectors.
- $\mathbf{b}$ is an m-vector with non-negative elements.

We can partition the problem as $\mathbf{x} = [\mathbf{x_B}, \mathbf{x_N}]$, $\mathbf{A} = [\mathbf{B}, \mathbf{N}]$, $\mathbf{c} = [\mathbf{c_B}, \mathbf{c_N}]$, where:

- $\mathbf{x_B}$ is the vector of basic variables
- $\mathbf{B}$ is the *basis matrix*, a nonsingular (i.e., it consists of m linearly dependent columns of $\mathbf{A}$) $m \times m$ matrix.
- $\mathbf{c_B}$ is the vector of cost coefficients for the basic variables.
- $\mathbf{x_N}$ is the vector of nonbasic variables

- $\mathbf{N}$ is the *nonbasic matrix*, a $m \times n - m$ matrix.
- $\mathbf{c_N}$ is the vector of cost coefficients for the basic variables.

The LP can then be written as:

$$\max\{\mathbf{c_B x_B} + \mathbf{c_N x_N} : \mathbf{B x_B} + \mathbf{N x_N} = \mathbf{b}, \mathbf{x_B}, \mathbf{x_N} \geq 0\}.$$

For the feasible region, we can write the system of equations as follows:

$$\mathbf{B x_B} + \mathbf{N x_N} = \mathbf{b}.$$

$$\mathbf{B x_B} = \mathbf{b} - \mathbf{N x_N}.$$

Premultiplying by $\mathbf{B}^{-1}$ yields:

$$\mathbf{x_B} = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N x_N}.$$

By setting $\mathbf{x_N} = 0$ and solving we (potentially) find a *basic feasible solution* to the system, which corresponds to an extreme point of the feasible region. Remember that the nonbasic variables $\mathbf{x_N} = 0$ represent the defining hyperplanes for a solution.

For any set of m variables, the result can be:

1. a basic feasible solution, $\mathbf{x_B} \geq 0$.
2. a basic infeasible solution, some $x \in \mathbf{x_B} \leq 0$.
3. a set of linearly dependent columns that does not span the m-space.

For this system there are possibly n choose m $\binom{n}{m}$ basic solutions, that is, the number of basic feasible solutions is bounded by $n!/m!(n-m)!$ from above.

4.2.4.1. The Simplex Algorithm

It is common practice to put an LP into a tableau. To do so, we first modify the objective function by bringing all the variables to the left-hand side, yielding the following tableau of LP data:

max	z	x_i	rhs
Row 0 (z)	1	$-c_i$	0
Rows 1-m	0	$\mathbf{a_i}$	$\mathbf{b}$

We are interested in tableaus that represent basic solutions, which have a special form; the columns of coefficients for the basis to form an identity matrix, which we can obtain using elementary row operations.

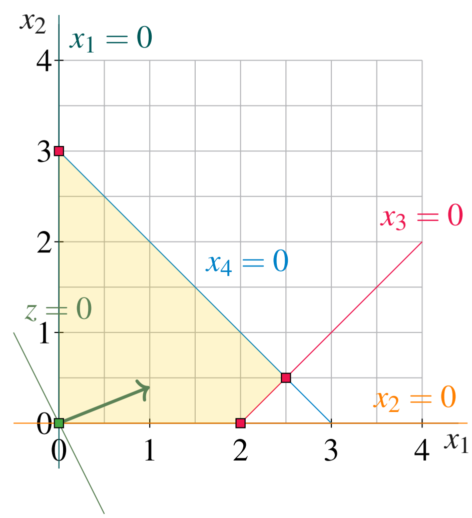
Consider the following LP:

$$\begin{aligned} \text{Max } & z = 2x_1 + x_2 \\ \text{s.t. } & x_1 - x_2 + x_3 = 2 \\ & x_1 + x_2 + x_4 = 3 \\ & x_1, x_2, x_3, x_4 \geq 0, \end{aligned}$$

where m=2, n=4, $\mathbf{A}$=$\begin{bmatrix} 1 & -1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$,
$\mathbf{a}_1$=$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $\mathbf{a}_2$=$\begin{bmatrix} -1 \\ 1 \end{bmatrix}$, $\mathbf{a}_3$=$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\mathbf{a}_4$=$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$,
$\mathbf{c}$=$\begin{bmatrix} 2 & 1 & 0 & 0 \end{bmatrix}$,
$\mathbf{x}$=$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T$, (T for transpose, $\mathbf{x}$ is a column vector),
and $\mathbf{b}$=$\begin{bmatrix} 2 \\ 3 \end{bmatrix}$.

The tableau and graph for this LP follow:

max	z	x_1	x_2	x_3	x_4	*rhs*
(z)	1	-2	-1	0	0	0
(x_3)	0	1	-1	1	0	2
(x_4)	0	1	1	0	1	3



Luckily, this tableau already represents a basis, which has basic variables $\mathbf{x_B} = [x_3\ x_4]^T$ (we can consider z as a basic variable of sorts to complete the identity matrix). Thus for this tableau we have $\mathbf{x_N} = [x_1\ x_2]^T$, $\mathbf{B}$=$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\mathbf{N}$=$\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$. This basic solution represents an extreme point if we set the nonbasic variables to zero, as we see in the graph of the LP in the nonbasic variable space.

Here the basic variable values are $x_3 = 2$ and $x_4 = 3$, and $z = 0$.

The simplex algorithm (mostly), tableau version:

1. Put the LP into a standard form tableau, find a set of basic variables that form a feasible basis, and modify the tableau to represent the basis using elementary row operations, we want the coefficients

of the basic variables to form an identity matrix.

2. Check optimality, for a maximization (minimization), if the Row 0 coefficients for the nonbasic variables are all nonnegative (nonpositive) then the basis is optimal.

3. If the basis is not optimal, then find an adjacent basis that improves the solution. This will involve swapping one of the basic variables with a nonbasic variable to form a new basis.

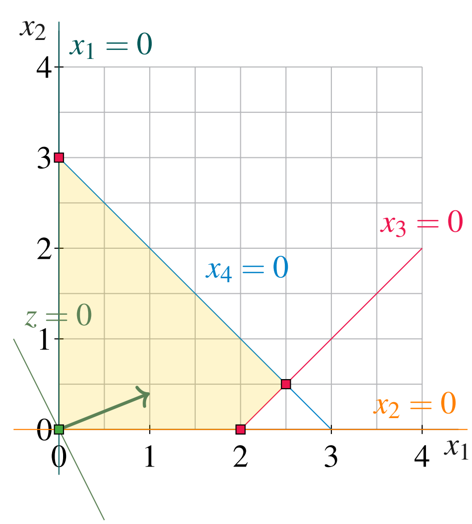
4. Select a nonbasic entering variable using Dantzig's rule, specifically, for a maximization (minimization) problem pick the nonbasic variable with the smallest negative (largest positive) reduced cost.

5. Select a variable to leave the basis. Conceptually, as we increase the entering variable's value from zero, the values of the basis variables should change, the basic variable that goes to zero first is the leaving variable. To find the leaving variable, use the ratio test. For each row 1-m having a positive coefficient in the entering variable column, divide the *rhs* by the entering variable's (positive) coefficient. The basic variable corresponding to row with the smallest ratio is the leaving variable.

6. Put the tableau into the proper form for the new basis using elementary row operations and go to Step 2.

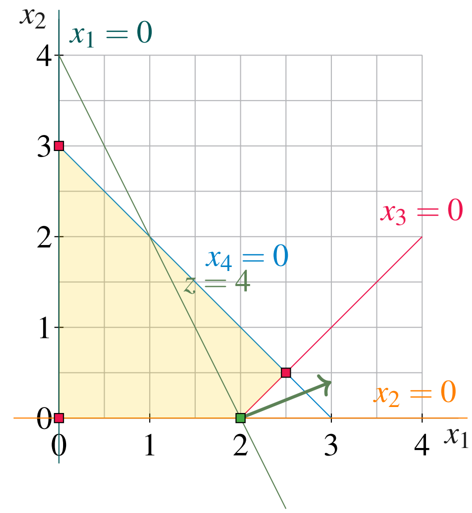
Consider the following example:

max	z	x_1	x_2	x_3	x_4	*rhs*
(z)	1	-2	-1	0	0	0
(x_3)	0	1	-1	1	0	2
(x_4)	0	1	1	0	1	3

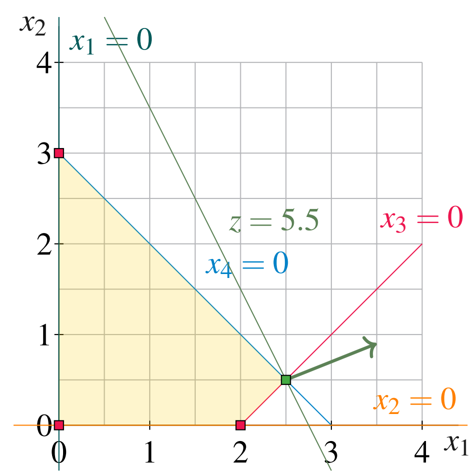


From the graph and tableau we can see that we are at the extreme point (0,0) and if we increase x_1 by one unit, the objective function (z-value) increases by 2, thus improving the solution. We can increase x_1 by 2 and still remain in the feasible region, moving to extreme point (2,0). Likewise, if we increase x_2 by one unit the objective function increases by 1, and we can increase x_2 by three and still remain in the feasible region, moving to extreme point (0,3). As x_2 increases the basic variable x_3 gets larger, while the basic variable x_4 gets smaller, so x_2 enters the basis and x_4 leaves.

max	z	x_1	x_2	x_3	x_4	rhs
z	1	0	-3	2	0	4
x_1	0	1	-1	1	0	2
x_4	0	0	2	-1	1	1



max	z	x_1	x_2	x_3	x_4	rhs
z	1	0	0	1/2	3/2	11/2
x_1	0	1	0	1/2	1/2	5/2
x_2	0	0	1	-1/2	1/2	1/2



Let's consider this algorithm, and what we know, and see if there are any missing parts, or other information we would find valuable.

- Unique optimal solution
- Multiple optimal solutions
- Unbounded optimal objective value
- Empty feasible region (an infeasible LP)

Tableau Formulas:

We can modify the tableau for a particular basis $\mathbf{B}$ using the following formulas:

	z	x_i	rhs
(z)	1	$\mathbf{c_B B^{-1} a_i - c_i}$	$\mathbf{c_B B^{-1} b}$
(x_B)	0	$\mathbf{B^{-1} a_i}$	$\mathbf{B^{-1} b}$

If we partition the variables, the formulas simplify as follows:

	z	x_B	x_N	rhs
z	1	$\mathbf{c_B B^{-1} B - c_B} = 0$	$\mathbf{c_B B^{-1} N - c_N}$	$\mathbf{c_B B^{-1} b}$
x_B	0	$\mathbf{B^{-1} B = I}$	$\mathbf{B^{-1} N}$	$\mathbf{B^{-1} b}$

The formulas for the coefficients for rows 1-m, that is $\mathbf{B^{-1} a_i}$ (or $\mathbf{B^{-1} A}$ for all the columns on the left-hand side) is fairly straight forward; Multiplying by $\mathbf{B}^{-1}$ is essentially the same as doing the elementary row operations required to get an identity matrix in the basic variable columns.

Now consider the formula $\mathbf{c_B B^{-1} a_i - c_i}$ for the row 0 coefficients. Where did this come from?

Consider an expanded basis matrix $\hat{\mathbf{B}}$, which includes the z-variable column and row, as follows: $\begin{bmatrix} 1 & -\mathbf{c_B} \\ 0 & \mathbf{B} \end{bmatrix}$, which yields $\hat{\mathbf{B}}^{-1}$ of $\begin{bmatrix} 1 & \mathbf{c_B B^{-1}} \\ 0 & \mathbf{B}^{-1} \end{bmatrix}$, and the column for x_i is $[-c_i, \mathbf{a_i}]^T$. Multiplying these yields $\begin{bmatrix} 1 & \mathbf{c_B B^{-1}} \\ 0 & \mathbf{B}^{-1} \end{bmatrix} \begin{bmatrix} -c_i \\ \mathbf{a_i} \end{bmatrix}$, which results in dot product of $[1, \mathbf{c_B B^{-1}}][-c_i, \mathbf{a_i}] = \mathbf{c_B B^{-1} a_i} - c_i$ for the first element of the resulting column vector.

For example, consider the following LP:

$$\begin{aligned} \text{Max} \quad & z = 2x_1 + 3x_2 \\ \text{s.t.} \quad & 1x_1 + 1x_2 \geq 2 \\ & 4x_1 + 6x_2 \leq 9 \\ & x_1, x_2 \geq 0. \end{aligned}$$

For this problem, if we have x_1 ad x_2 as the basic variables, then

$$\hat{\mathbf{B}} = \begin{bmatrix} 1 & -2 & -3 \\ 0 & 1 & 1 \\ 0 & 4 & 6 \end{bmatrix} \text{ and } \hat{\mathbf{B}}^{-1} = \begin{bmatrix} 1 & 0 & 1/2 \\ 0 & 3 & -1/2 \\ 0 & -2 & 1/2 \end{bmatrix},$$

and the column for x_1 is $[-2, 1, 4]^T$.

When we multiply $\hat{\mathbf{B}}^{-1}$ and $[-2,1,4]^T$ we get

$$\begin{bmatrix} 1 & 0 & 1/2 & -2 & & 0 \\ 0 & 3 & -1/2 & 1 & = & 1 \\ 0 & -2 & 1/2 & 4 & & 0 \end{bmatrix}$$

The simplex algorithm again:

1. Put the LP into a standard form tableau, find a feasible basis $\mathbf{B}$ and modify the tableau using $\mathbf{B}^{-1}$ and the tableau formulas.
2. Check optimality, if $\mathbf{c_B B^{-1} a_i} - \mathbf{c_i} \geq (\leq)\ 0$ for $i : x_i \in \mathbf{x_N}$ for a maximization (minimization) problem, then the current basic solution is optimal. Stop.
3. Select an entering nonbasic variable using Dantzig's rule, specifically, entering variable x_i, where $i =$
   min (max)$_i$ $\{\mathbf{c_B B^{-1} a_i} - \mathbf{c_i} < (>)\ 0 : (i : x_i \in \mathbf{x_N})\}$ for a maximization (minimization) problem.
4. Select a variable to leave the basis using the ratio test. For entering variable x_i the leaving variable is the basic variable corresponding to row j, where :
   $\min_j\{[\mathbf{B}^{-1}\mathbf{b}]_j/[\mathbf{B}^{-1}\mathbf{a_i}]_j,\ (j : j = 1,\cdots,m,\ [\mathbf{B}^{-1}\mathbf{a_i}]_j > 0)\}$.
5. Put the tableau into the proper form for the new basis and go to Step 2.

Finding an Initial BFS When a basic feasible solution is not apparent, we an produce one using *artificial variables*. This *artificial* basis is undesirable from the perspective of the original problem, we do not want the artificial variables in our solution, so we penalize them in the objective function, and allow the simplex algorithm to drive them to zero (if possible) and out of the basis. There are two such methods, the **Big M method** and the **Two-phase method**, which we illustrate below:

Solve the following LP using the Big M Method and the simplex algorithm:

$$\begin{aligned} max\ \ & z = 9x_1 + 6x_2 \\ s.t.\ \ & 3x_1 + 3x_2 \leq 9 \\ & 2x_1 - 2x_2 \geq 3 \\ & 2x_1 + 2x_2 \geq 4 \\ & x_1, x_2 \geq 0. \end{aligned}$$

Here is the LP is transformed into standard form by using slack variables x_3, x_4, and x_5, with the required artificial variables x_6 and x_7, which allow us to easily find an initial basic feasible solution (to the artificial problem).

$$\begin{aligned} max \quad & z_a = 9x_1 + 6x_2 - Mx_6 - Mx_7 \\ s.t. \quad & 3x_1 + 3x_2 + x_3 = 9 \\ & 2x_1 - 2x_2 - x_4 + x_6 = 3 \\ & 2x_1 + 2x_2 - x_5 + x_7 = 4 \\ & x_i \geq 0, \ \ i = 1, \cdots, 7. \end{aligned}$$

z	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS	ratio
1	-9	-6	0	0	0	M	M	0	
0	3	3	1	0	0	0	0	9	
0	2	-2	0	-1	0	1	0	3	
0	2	2	0	0	-1	0	1	4	

This tableau is not in the correct form, it does not represent a basis, the columns for the artificial variables need to be adjusted.

z	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS	ratio
1	-9 - 4M	-6	0	M	M	0	0	-7M	
0	3	3	1	0	0	0	0	9	3
0	2	-2	0	-1	0	1	0	3	3/2
0	2	2	0	0	-1	0	1	4	2

The current solution is not optimal, so x_1 enters the basis, and by the ratio test, x_6 (an artificial variable) leaves the basis.

z	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS	ratio
1	0	-15 -4M	0	-9/2 -M	M	9/2 + 2M	0	27/2 -M	
0	0	6	1	3/2	0	-3/2	0	3/2	3/4
0	1	-1	0	-1/2	0	1/2	0	3/2	-
0	0	4	0	1	-1	-1	1	1	1/4

The current solution is not optimal, so x_2 enters the basis, and by the ratio test, x_7 (an artificial variable) leaves the basis.

z	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS	ratio
1	0	0	0	-3/4	-15/4	-	-	17 1/4	
0	0	0	1	0	3/2	0	-3/2	3	-
0	1	0	0	-1/4	-1/4	1/2	1/4	7/4	-
0	0	1	0	1/4	-1/4	-1/4	1/4	1/4	1

The current solution is not optimal, so x_4 enters the basis, and by the ratio test, x_2 leaves the basis.

z	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS	ratio
1	0	3	0	0	-9/2	-	-	18	
0	0	0	1	0	3/2	0	-3/2	3	-
0	1	1	0	0	-1/2	0	1/2	2	-
0	0	4	0	1	-1	-1	1	1	1

The current solution is not optimal, so x_5 enters the basis, and by the ratio test, x_3 leaves the basis.

z	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS	ratio
1	0	3	3	0	0	-	-	27	
0	0	0	2/3	0	1	0	-1	2	
0	1	1	1/3	0	0	0	0	3	
0	0	4	2/3	1	0	-1	0	3	

The current solution is optimal!

Solve the following LP using the Two-phase Method and Simplex Algorithm.

$$\begin{aligned} max \;\; & z = 2x_1 + 3x_2 \\ s.t. \;\; & 3x_1 + 3x_2 \geq 6 \\ & 2x_1 - 2x_2 \leq 2 \\ & -3x_1 + 3x_2 \leq 6 \\ & x_1, x_2 \geq 0. \end{aligned}$$

Here is first phase LP (in standard form), where x_3, x_4, and x_5 are slack variables, and x_6 is an artificial variable.

$$\begin{aligned} min \;\; & z_a = x_6 \\ s.t. \;\; & 3x_1 + 3x_2 - x_3 + x_6 = 6 \\ & 2x_1 - 2x_2 + x_4 = 2 \\ & -3x_1 + 3x_2 + x_5 = 6 \\ & x_i \geq 0, \;\; i = 1, \cdots, 6. \end{aligned}$$

Next, we put the LP into a tableau, which, still is not in the right form for our basic variables (x_6, x_4, and x_5).

z	x_1	x_2	x_3	x_4	x_5	x_6	RHS	ratio
1	0	0	0	0	0	-1	0	
0	3	3	-1	0	0	1	6	
0	2	-2	0	1	0	0	2	
0	-3	3	0	0	1	0	6	

To remedy this, we use row operation to modify the row 0 coefficients, yielding the following:

z	x_1	x_2	x_3	x_4	x_5	x_6	RHS	ratio
1	3	3	-1	0	0	0	6	
0	3	3	-1	0	0	1	6	2
0	2	-2	0	1	0	0	2	-
0	-3	3	0	0	1	0	6	2

The current solution is not optimal, either x_1 or x_2 can enter the basis, let's choose x_2. Then by the ratio test, either x_6 (an artificial variable) or x_5 (a slack variable) can leaves the basis. Let's choose x_6.

z	x_1	x_2	x_3	x_4	x_5	x_6	RHS	ratio
1	0	0	0	0	0	-1	0	
0	1	1	-1/3	0	0	1/3	2	
0	4	0	-2/3	1	0	2/3	6	
0	-6	0	1	0	1	-1	0	

The current solution is optimal, so we end the first phase with a basic feasible solution to the original problem, with x_2, x_4, and x_5 as the basic variables. Now we provide a new row zero that corresponds to the original problem.

z	x_1	x_2	x_3	x_4	x_5	x_6	RHS	ratio
1	1	0	-1	0	0	0	6	
0	1	1	-1/3	0	0	1/3	2	
0	4	0	-2/3	1	0	2/3	6	
0	-6	0	1	0	1	-1	0	

z	x_1	x_2	x_3	x_4	x_5	x_6	RHS	ratio
1	-5	0	0	0	1	-1	6	
0	-1	1	0	0	1/3	0	2	
0	0	0	0	1	2/3	0	6	
0	-6	0	1	0	1	-1	0	

From this tableau we can see that the LP is unbounded and an extreme point is [0, 2, 0, 6,0] and an extreme direction is [1, 1, 6, 0, 0].

Degeneracy and the Simplex Algorithm

Degeneracy must be considered in the simplex algorithm, as it causes some trouble. For instance, it might mislead us into thinking there are multiple optimal solutions, or provide faulty insight. Further, the algorithm as described can *cycle*, that is, remain on a degenerate extreme point repeatedly cycling through a subset of bases that represent that point, never leaving.

min

z	x_1	x_2	x_3	x_4	x_5	x_6	x_7	rhs
1	0	0	0	3/4	-20	1/2	-6	0
0	1	0	0	1/4	-8	-1	9	0
0	0	1	0	1 /2	-12	-1/2	3	0
0	0	0	1	0	0	1	0	1

Solve the following LP using the Simplex Algorithm:

$$\begin{aligned} \max \quad & z = 40x_1 + 30x_2 \\ s.t. \quad & 6x_1 + 4x_2 \leq 40 \\ & 4x_1 + 2x_2 \leq 20 \\ & x_1, x_2 \geq 0. \end{aligned}$$

By adding slack variables, we have the following tableau. Luckily, this tableau represents a basis, where BV=$\{s_1, s_2\}$, but by inspecting the row 0 (objective function row) coefficients, we can see that this is not

z	x_1	x_2	s_1	s_2	RHS
1	-40	-30	0	0	0
0	6	4	1	0	40
0	4	2	0	1	20

optimal. By Dantzig's Rule, we enter x_1 into the basis, and by the ratio test we see that s_2 leaves the basis. By performing elementary row operations, we obtain the following tableau for the new basis BV=$\{s_1, x_1\}$.

z	x_1	x_2	s_1	s_2	RHS
1	0	-10	0	10	200
0	0	1	1	-3/2	10
0	1	1/2	0	1/4	5

This tableau is not optimal, entering x_2 into the basis can improve the objective function value. The basic variables s_1 and x_1 tie in the ration test. If we have x_1 leave the basis, we get the following tableau (BV=$\{s_1, x_2\}$).

z	x_1	x_2	s_1	s_2	RHS
1	20	0	0	15	300
0	-2	0	1	-2	0
0	2	1	0	1/2	10

This is an optimal tableau, with an objective function value of 300, If instead of x_1 leaving the basis, suppose s_1 left, this would lead to the following tableau (BV=$\{x_2, x_1\}$).

z	x_1	x_2	s_1	s_2	RHS
1	0	0	10	-5	300
0	0	1	1	-3/2	10
0	1	0	-1/2	1	0

This tableau does not look optimal, yet the objective function value is the same as the optimal solution's. This occurs because the optimal extreme point is a degenerate.

4.2.4.2. Dual Simplex Algorithm

The dual simplex algorithm is essentially performing the simplex algorithm, on the dual problem, on the primal tableau. Remember, the Simplex algorithm, in relation to the KKT conditions, maintains primal feasibility, complementary slackness, and strives for dual feasibility. The dual simplex algorithm maintains dual feasibility, complementary slackness, and strives for primal feasibility.

1. Pick the row with the smallest $\bar{b}_i$, where $\bar{b}_i < 0$ ($\bar{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b}$), this corresponds to the leaving variable.
2. Pick a column with the minimum $\{|z_j - c_j / y_{ij}| : y_{ij} < 0\}$, this corresponds to the entering variable.
3. Pivot, and repeat until primal feasibility is achieved.

4.2.4.3. Primal-Dual Algorithm

The primal dual algorithm is another method for solving LPs. This algorithm starts with a feasible dual solution (not necessarily basic) and searches for a primal feasible solution will maintaining complementary slackness between the primal and dual solutions. Consider the following primal dual pair:

$$(P) : \min\{\mathbf{cx} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq 0\}$$
$$(D) : \max\{\mathbf{wb} : \mathbf{wA} \leq \mathbf{c}, \mathbf{w}\ urs\}.$$

To solve P using the primal-dual algorithm, first find a feasible solution to D, it does not necessarily have to be basic, and form the following restricted primal problem:

$$(P_R) : \min\{z_R = 1\mathbf{x^a} : \mathbf{a_j}x_j + \mathbf{x^a} = \mathbf{b}, x_j \geq 0,\ j \in Q,\ \mathbf{x^a} \geq 0\},$$

where $Q = \{j : wa_j = c_j\}$, that is, Q is the set of indexes for binding dual constraints, and $x_j, j \in Q$ are the primal variables that can be non-zero given the current dual solution and complementary slackness. The vector $\mathbf{x^a}$ is a vector of artificial variables; this problem looks very similar to the phase 1 problem in the two-phase method. The objective is the same, to find a basic feasible primal solution, but here we use a *restricted* or limited number of primal variables, those with indexes in set Q.

Solve P_R (using simplex) and if $z_R = 0$, stop, the solution is optimal for P because we have satisfied the KKT conditions, else let $(v)^*$ be the corresponding optimal dual solution D_R.

$$D_R : \max\{\mathbf{vb} : \mathbf{va_j} \leq 0,\ i \in Q, \mathbf{v} \leq 1, \mathbf{v}\ urs\}.$$

Note that for each $j \in Q$, $\mathbf{v}^*\mathbf{a_j} \leq 0$. For $i \notin Q$ calculate $\mathbf{v}^*\mathbf{a_i}$, if $\mathbf{v}^*\mathbf{a_i} > 0$ then x_i can be added to the restricted primal to improve z_R. To get x_i into Q we must modify the original dual solution $\mathbf{w}$, first we calculate θ.

$$\theta = \min_{i \notin Q}\{|(\mathbf{wa_i} - c_i)|/\mathbf{v}^*\mathbf{a_i} : \mathbf{v}^*\mathbf{a_i} > 0\} > 0$$

We take the absolute value of $\mathbf{wa_i} - c_i)$ because if the primal problem is a minimization, this term will always be non-positive, for a primal maximization this will always be non-negative (thus the absolute value is not needed).

We then replace $\mathbf{w}$ by $\mathbf{w} + \theta\mathbf{v}^*$. We use this θ-step like a ratio test, changing the current dual solution such that we can enter a new primal variable into the restricted primal, one that will improve the solution (and move us closer to feasibility), while maintaining dual feasibility and complementary slackness.

If we think about this algorithm on a tableau, we get the following formulas, where the z_P row corresponds to our dual solution, and defines the set Q, and simplex is performed on on the restricted problem (using row z_P^R for the objective function row, and using column defined by Q).

	z	$\mathbf{x}$	$\mathbf{x^s}$	$\mathbf{x^a}$	rhs
z_P	1	$\mathbf{wa_i} - c_i$	$\mathbf{wa_i}$	0	0
z_P^R	1	$\mathbf{va_i}$	$\mathbf{va_i}$	$\mathbf{va_i} - 1$	0
	0	$\mathbf{B^{-1}a_i}$	$\mathbf{B^{-1}a_i}$	$\mathbf{B^{-1}a_i}$	$\mathbf{B^{-1}b}$

Example 4.2: Primal-dual algorithm

Consider again the following LP and solve using the primal-dual algorithm, using a starting feasible dual solution of $\mathbf{w} = [10,0,0]$.

Solution.

$$\begin{aligned} \max \quad & 18x_1 + 16x_2 + 10x_3 \\ s.t. \quad & 2x_1 + 2x_2 + 1x_3 + x_4^s = 21 \;\; (w_1) \\ & 3x_1 + 2x_2 + 2x_3 + x_5^s = 23 \;\; (w_2) \\ & 1x_1 + 2x_2 + 1x_3 + x_6 = 17 \;\; (w_3) \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0. \end{aligned}$$

$$\begin{aligned} D1: \quad \min \quad & 21w_1 + 23w_2 + 17w_3 \\ s.t. \quad & 2w_1 + 3w_2 + 1w_3 \geq 18 \;\; (x_1) \\ & 2w_1 + 2w_2 + 2w_3 \geq 16 \;\; (x_2) \\ & 1w_1 + 2w_2 + 1w_3 \geq 10 \;\; (x_3) \\ & w_1, w_2, w_3 \geq 0. \end{aligned}$$

Use the modified tableau method, and write Q and the restricted problem for each step. Be able to explain the complete process.

For $\mathbf{w} = [10,0,0]$ we have $Q = \{3,5,6\}$

	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	x_7^a	rhs
z	2	4	0	10	0	0	0	210
z_R	0	0	0	0	0	0	-1	0
	2	2	1	1	0	0	1	21
	3	2	2	0	1	0	0	23
	1	2	1	0	0	1	0	17

We need to make a minor adjustment to the artificial variable column.

	z_P	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	x_7^a	rhs
z_P	1	2	4	0	10	0	0	0	210
z_R	1	2	2	1	1	0	0	0	21
	0	2	2	1	1	0	0	1	21
	0	3	2	2	0	1	0	0	23
	0	1	2	1	0	0	1	0	17

x_3 enters the restricted basis and x_5^s leaves the restricted basis, resulting in the following tableau:

	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	x_7^a	rhs
z	2	4	0	10	0	0	0	210
z_R	1/2	1	0	1	-1/2	0	0	19/2
	1/2	1	0	1	-1/2	0	1	19/2
	3/2	1	1	0	1/2	0	0	23/2
	-1/2	1	0	0	-1/2	1	0	11/2

To find Θ we take the minimum of 4/1 and 2/(1/2), which are equal, and thus $\Theta = 4$, using this we adjust the z-row, yielding the following tableau. Notice that this operation ensures that the z-row remains non-negative. Now $Q = \{1,2,3,6\}$.

	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	x_7^a	rhs
z	0	0	0	6	2	0	0	172
z_R	1/2	1	0	1	-1/2	0	0	19/2
	1/2	1	0	1	-1/2	0	1	19/2
	3/2	1	1	0	1/2	0	0	23/2
	-1/2	1	0	0	-1/2	1	0	11/2

x_2 enters the basis (of the restricted problem) and x_6 leaves.

	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	x_7^a	rhs
z	0	0	0	6	2	0	0	172
z_R	1	0	0	1	0	-1	0	4
	1	0	0	1	0	-1	1	4
	2	0	1	0	1	-1	0	6
	-1/2	1	0	0	-1/2	1	0	11/2

This is not an optimal solution, so now x_1 enters the restricted basis and x_3 leaves.

	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	x_7^a	rhs
z	0	0	0	6	2	0	0	172
z_R	0	0	-1/2	1	-1/2	-1/2	0	1
	0	0	-1/2	1	-1/2	-1/2	1	1
	1	0	1/2	0	1/2	-1/2	0	3
	0	1	1/4	0	-1/4	3/4	0	7

Here $\Theta = 6$, which yields the following, having Now $Q = \{1,2,4\}$; we can see that x_4 enters and a_1 leaves, which will change the z_R-row, this makes the restricted primal optimal, but does not change any other rows, and thus this is the optimal solution.

	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	x_7^a	rhs
z	0	0	3	0	5	3	0	166
z_R	0	0	-1/2	1	-1/2	-1/2	0	1
	0	0	-1/2	1	-1/2	-1/2	1	1
	1	0	1/2	0	1/2	-1/2	0	3
	0	1	1/4	0	-1/4	3/4	0	7

♠

4.2.5. Sensitivity Analysis

Consider an arbitrary LP, which we will call the primal (P):

$$(P) : \max\{\mathbf{cx} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq 0\},$$

where $\mathbf{A}$ is an $m \times n$ matrix, and $\mathbf{x}$ is a n element column vector. Every prmal LP has a related LP, which we call the dual, the dual of (P) is:

$$(D) : \min\{\mathbf{wb} : \mathbf{wA} \geq \mathbf{c}, \mathbf{w} \geq 0\}.$$

If an LP has a different form from P, we can convert it to the above form to find the dual, or use the rules in the following table:

max $\mathbf{cx}$:	min $\mathbf{wb}$:
$\mathbf{a}_{1*}\mathbf{x} \leq b_1 \ (w_1 \geq 0)$	$\mathbf{wa}_{*1} \geq c_1 \ (x_1 \geq 0)$
$\mathbf{a}_{2*}\mathbf{x} = b_2 \ (w_2 \ urs)$	$\mathbf{wa}_{*2} = c_2 \ (x_2 \ urs)$
$\mathbf{a}_{3*}\mathbf{x} \geq b_3 \ (w_3 \leq 0)$	$\mathbf{wa}_{*3} \leq c_3 \ (x_3 \leq 0)$
$\vdots$	$\vdots$
$x_1 \geq 0, x_2 \ urs, x_3 \leq 0, \cdots$	$w_1 \geq 0, w_2 \ urs, w_3 \leq 0, \cdots$

Define $\mathbf{w} = \mathbf{c}_B\mathbf{B}^{-1}$ as the vector of *shadow prices*, where w_i represents the change in the objective function value caused by a unit change to the associated b_i parameter (i.e., increasing the amount of resource i by one unit, see dual objective function).

Some observations:

- The dual of D is P.
- Each primal constraint has an associated dual variable (w_i) and each dual constraint has an associated primal variable (x_i).
- When the primal is a maximization, the dual is a minimization, and vice versa.

Consider the following primal tableau (where z_p is the primal objective function value) for $(P): \ Max\ \{\mathbf{cx} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq 0\}$.

	z_P	x_i	rhs
z_P	1	$\mathbf{c_B B^{-1} a_i} - c_i$	$\mathbf{c_B B^{-1} b}$
BV	0	$\mathbf{B^{-1} a_i}$	$\mathbf{B^{-1} b}$

Observe that if a basis for P is optimal, then the row zero coefficients for the variables are greater than, or equal to, zero, that is, $c_B B^{-1} a_i - c_i \geq 0$ for each x_i (if the variable is a slack, this simplifies to $c_B B^{-1} \geq 0$).

Substituting $w = c_B B^{-1}$ we get $\mathbf{wA} \geq \mathbf{c}, \mathbf{w} \geq 0$ which corresponds to dual feasibility.

$$(D) : \min\{\mathbf{wb} : \mathbf{wA} \geq \mathbf{c}, \mathbf{w} \geq 0\}.$$

Weak Duality Property

If $\mathbf{x}$ and $\mathbf{w}$ are feasible solutions to P and D, respectively, then $\mathbf{cx} \leq \mathbf{wAx} \leq \mathbf{wb}$.

$$(P) : \max\{\mathbf{cx} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq 0\}.$$

$$(D) : \min\{\mathbf{wb} : \mathbf{wA} \geq \mathbf{c}, \mathbf{w} \geq 0\}.$$

This implies that the objective function value for a feasible solution to P is a lower bound on the objective function value for the optimal solution to D, and the objective function value for a feasible solution to D is an upper bound on the objective function value for the optimal solution to P.

Thus if the objective function values are equal, i.e., $\mathbf{cx} = \mathbf{wb}$, then the solutions $\mathbf{x}$ and $\mathbf{w}$ are optimal.

Theorem 4.3: Fundamental Theorem of Duality

or problems P and D (i.e., any primal dual set) exactly one of the following is true:

1. *Both have optimal solutions $\mathbf{x}$ and $\mathbf{w}$ where $\mathbf{cx} = \mathbf{wb}$.*
2. *One problem is unbounded (i.e., the objective function value can become arbitrarily large for a maximization, or arbitrarily small for a minimization), and the other is infeasible.*
3. *Both are infeasible.*

Theorem 4.4: Farkás Lemma

onsider the following two systems:

1. $\mathbf{Ax} \geq 0$, $\mathbf{cx} < 0$.
2. $\mathbf{wA} = \mathbf{c}$, $\mathbf{w} \geq 0$.

Exactly one of these systems has a solution.

Suppose system 1 has x as a solution:

- If $\mathbf{w}$ were a solution to system 2, then post-multiplying each side of $\mathbf{wA} = \mathbf{c}$ by $\mathbf{x}$ would yield $\mathbf{wAx} = \mathbf{cx}$.
- Since $\mathbf{Ax} \geq 0$ and $\mathbf{w} \geq 0$, this implies that $\mathbf{cx} \geq 0$, which violates $\mathbf{cx} < 0$.
- Thus we show that if system 1 has a solution, system 2 cannot have one.

Suppose system 1 has no solution:

- Consider the following LP: $\min\{\mathbf{cx} : \mathbf{Ax} \geq 0\}$.
- The optimal solution is $\mathbf{cx} = 0$ and $\mathbf{x} = 0$.
- The LP in standard form (substitute $\mathbf{x} = \mathbf{x}' - \mathbf{x}''$, $\mathbf{x}' \geq 0$ and $\mathbf{x}'' \geq 0$ and add $\mathbf{x^s} \geq 0$) follows:
$$\min\{\mathbf{cx}' - \mathbf{cx}'' : \mathbf{Ax}' - \mathbf{Ax}'' - \mathbf{x^s} = 0, \mathbf{x}', \mathbf{x}'', \mathbf{x^s} \geq 0\}$$
- $\mathbf{x}' = 0$, $\mathbf{x}'' = 0$, $\mathbf{x^s} = 0$ is an optimal extreme point solution.
- Using $\mathbf{x^s}$ as an initial feasible basis, solve with the simplex algorithm (with cycling prevention) to find a basis where $\mathbf{c_B B^{-1} a_i} - c_i \leq 0$ for all variables. Define $\mathbf{w} = \mathbf{c_B B^{-1}}$.
- This yields $\mathbf{wA} - \mathbf{c} \leq 0$, $-\mathbf{wA} + \mathbf{c} \leq 0$, $-\mathbf{w} \leq 0\}$, from the columns for variables $\mathbf{x}'$, $\mathbf{x}''$, $\mathbf{x^s}$, respectively. Thus, $\mathbf{w} \geq 0$ and $\mathbf{wA} = \mathbf{c}$, and system 2 has a solution.

Karush-Kuhn-Tucker (KKT) Conditions

$$(P) : \max\{\mathbf{cx} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq 0\}.$$
$$(D) : \min\{\mathbf{wb} : \mathbf{wA} \geq \mathbf{c}, \mathbf{w} \geq 0\}.$$

For problems P and D, with solutions $\mathbf{x}$ and $\mathbf{w}$, respectively, we have the following conditions, which for LPs are necessary and sufficient conditions for optimality:

1. $\mathbf{Ax} \leq \mathbf{b}$, $\mathbf{x} \geq 0$ (primal feasibility).
2. $\mathbf{wA} \geq \mathbf{c}$, $\mathbf{w} \geq 0$ (dual feasibility).
3. $\mathbf{w}(\mathbf{Ax} - \mathbf{b}) = 0$ and $\mathbf{x}(\mathbf{c} - \mathbf{wA}) = 0$ (complementary slackness).

Note we can rewrite the third condition as $\mathbf{w}(\mathbf{Ax} - \mathbf{b}) = \mathbf{wx^s} = 0$ and $\mathbf{x}(c - \mathbf{wA}) = \mathbf{xw^s} = 0$, where $\mathbf{x^s}$ and $\mathbf{w^s}$ are the slack variables for the primal and dual problems, respectively.

Why do the KKT conditions hold?

Suppose that the LP $\min\{\mathbf{cx} : \mathbf{Ax} \geq \mathbf{b}, \mathbf{x} \geq 0\}$ has an optimal solution $\mathbf{x}^*$ (the dual is $\max\{\mathbf{wb} : \mathbf{wA} \leq \mathbf{c}, \mathbf{w} \geq 0\}$).

- Since $\mathbf{x}^*$ is optimal there is no direction $\mathbf{d}$ such that $\mathbf{c}(\mathbf{x}^* + \lambda \mathbf{d}) < \mathbf{c}\mathbf{x}^*$, $\mathbf{A}(\mathbf{x}^* + \lambda \mathbf{d}) \geq \mathbf{b}$, and $\mathbf{x}^* + \lambda \mathbf{d} \geq 0$ for $\lambda > 0$.
- Let $\mathbf{G}\mathbf{x} \geq \mathbf{g}$ be the binding inequalities in $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ and $\mathbf{x} \geq 0$ for solution $\mathbf{x}^*$ that is, $\mathbf{G}\mathbf{x}^* = \mathbf{g}$.
- Based on the optimality of $\mathbf{x}^*$, there is no direction $\mathbf{d}$ at $\mathbf{x}^*$ such that $\mathbf{c}\mathbf{d} < 0$ and $\mathbf{G}\mathbf{d} \geq 0$ (else we could improve the solution).
- Based on Farka's Lemma, if the system $\mathbf{c}\mathbf{d} < 0$, $\mathbf{G}\mathbf{d} \geq 0$ does not have a solution, the system $\mathbf{w}\mathbf{G} = \mathbf{c}$, $\mathbf{w} \geq 0$ must have a solution.
- $\mathbf{G}$ is composed of rows from $\mathbf{A}$ where $\mathbf{a_{i*}}\mathbf{x}^* = b_i$ and vectors $\mathbf{e_i}$ for any $x_i^* = 0$.
- We can divide the $\mathbf{w}$ into two sets:
  - $\{w_i,\ i : \mathbf{a_{i*}}\mathbf{x}^* = b_i\}$ - those corresponding to the binding functional constraints in the primal.
  - $\{w_i^s,\ j : x_i^* = 0\}$ - those corresponding to the binding non-negativity constraints in the primal.
- Thus $\mathbf{G}$ has the columns $\mathbf{a_{i*}^T}$ for w_i and e_i^T for w_i^s.
- Since $\mathbf{w}\mathbf{G} = \mathbf{c}$, $\mathbf{w} \geq 0$ must have a solution, this solution is feasible for $\mathbf{w}\mathbf{A} \leq \mathbf{c}, \mathbf{w} \geq 0$ where w_i^s are added slacks. Thus, $\mathbf{G}$ is missing some columns from $\mathbf{A}$ (and thus some w variables) and some slack variables if $\mathbf{w}\mathbf{A} \leq \mathbf{c}, \mathbf{w} \geq 0$ were put into standard form, but those are not needed for feasibility based on the result, and thus can be thought of as set to zero, giving us complementary slackness.

Example:

Example 4.5: Production LP

Consider a production LP (the primal P) where the variables represent the amount of three products to produce, using three resources, represented by the functional constraints. In standard form P and D have x_4^s, x_5^s, x_6^s and w_4^s, w_5^s, w_6^s as slack variables, respectively.

Solution. Decision variables:

x_i : number of units of product i to produce, $\forall i = \{1,\ 2,\ 3\}$.

$$
\begin{aligned}
(P): \max \ & z_P = 18x_1 + 16x_2 + 10x_3 \\
s.t. \ & 2x_1 + 2x_2 + 1x_3 + x_4^s = 21 \ \ (w_1) \\
& 3x_1 + 2x_2 + 2x_3 + x_5^s = 23 \ \ (w_2) \\
& 1x_1 + 2x_2 + 1x_3 + x_6^s = 17 \ \ (w_3) \\
& x_1, x_2, x_3, x_4^s, x_5^s, x_6^s \geq 0.
\end{aligned}
$$

$$
\begin{aligned}
(D): \min \ & z_D = 21w_1 + 23w_2 + 17w_3 \\
s.t. \ & 2w_1 + 3w_2 + 1w_3 \geq 18 \ \ (x_1) \\
& 2w_1 + 2w_2 + 2w_3 \geq 16 \ \ (x_2) \\
& 1w_1 + 2w_2 + 1w_3 \geq 10 \ \ (x_3) \\
& 1w_1 \geq 0 \\
& 1w_2 \geq 0 \\
& 1w_3 \geq 0 \\
& w_1, w_2, w_3 \ urs.
\end{aligned}
$$

Decision variables:
w_i : unit selling price for resource i, $\forall i = \{1,\ 2,\ 3\}$.

$$
\begin{aligned}
(D): \min \ & z_D = 21w_1 + 23w_2 + 17w_3 : \\
& 2w_1 + 3w_2 + 1w_3 - w_4^s = 18 \ \ (x_1) \\
& 2w_1 + 2w_2 + 2w_3 - w_5^s = 16 \ \ (x_2) \\
& 1w_1 + 2w_2 + 1w_3 - w_6^s = 10 \ \ (x_3) \\
& w_1, w_2, w_3, w_4^s, w_5^s, w_6^s \geq 0.
\end{aligned}
$$

The initial basic feasible tableau for the primal, i.e., having the slack variables form the basis, follows:

P : max	z_P	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	rhs
z_P	1	-18	-16	-10	0	0	0	0
x_4^s	0	2	2	1	1	0	0	21
x_5^s	0	3	2	2	0	1	0	23
x_6^s	0	1	2	1	0	0	1	17

$$x_1, x_2, x_3 = 0,\ x_4^s = 21,\ x_5^s = 23,\ x_6^s = 17\ z_P = 0$$

The following dual tableau **conforms with the primal tableau through complementary slackness**.

D : min	z_D	w_1	w_2	w_3	w_4^s	w_5^s	w_6^s	rhs
z_D	1	-21	-23	-17	0	0	0	0
w_4^s	0	-2	-3	-1	1	0	0	-18
w_5^s	0	-2	-2	-2	0	1	0	-16
w_6^s	0	-1	-2	-1	0	0	1	-10

$$w_1, w_2, w_3 = 0,\ w_4^s = -18,\ w_5^s = -16,\ w_6^s = -10\ z_D = 0$$

Complementary slackness: $w_1 x_4^s = 0,\ w_2 x_5^s = 0,\ w_3 x_6^s = 0,\ x_1 w_4^s = 0,\ x_2 w_5^s = 0,\ x_3 w_6^s = 0.$

- If a primal variable is basic, then its corresponding dual variable must be nonbasic, and vise versa.
- The primal is suboptimal, and the dual tableau has a basic infeasible solution.
- Row 0 of the primal tableau has dual variable values in the corresponding primal variable columns.

The primal basis is not optimal, so enter x_1 into the basis, and remove x_5^s, which yields:

P: Max	z_P	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	rhs
z_P	1	0	-4	2	0	6	0	138
x_4^s	0	0	2/3	-1/3	1	-2/3	0	17/3
x_1	0	1	2/3	2/3	0	1/3	0	23/3
x_6^s	0	0	4/3	1/3	0	-1/3	1	28/3

D: Min	z_D	w_1	w_2	w_3	w_4^s	w_5^s	w_6^s	rhs
z_D	1	-17/3	0	-28/3	-23/3	0	0	138
w_2	0	2/3	1	1/3	-1/3	0	0	6
w_5^s	0	-2/3	0	-4/3	-2/3	1	0	-4
w_6^s	0	1/3	0	-1/3	-2/3	0	1	2

The primal tableau does not represent an optimal basic solution, and the dual tableau does not represent a feasible basic solution.

Using Dantzig's rule, we enter x_2 into the basis, and using the ratio test we find that x_6^s leaves the basis. This change in basis yields the following tableau:

P: Max	z_P	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	rhs
z_P	1	0	0	3	0	5	3	166
x_4^s	0	0	0	-1/2	1	-1/2	-1/2	1
x_1	0	1	0	1/2	0	1/2	-1/2	3
x_2	0	0	1	1/4	0	-1/4	3/4	7

D: Min	z_D	w_1	w_2	w_3	w_4^s	w_5^s	w_6^s	rhs
z_D	1	-1	0	0	-3	-7	0	166
w_2	0	1/2	1	0	-1/2	1/4	0	5
w_3	0	1/2	0	1	1/2	-3/4	0	3
w_6^s	0	1/2	0	0	-1/2	-1/4	1	3

Decision variables:
x_i : number of units of product i to produce, $\forall i = \{1,\ 2,\ 3\}$.

$$
\begin{aligned}
(P) : \max\ \ z_P = 18x_1 + 16x_2 + 10x_3 :& \\
2x_1 + 2x_2 + 1x_3 + x_4^s = 21 &\ \ (w_1) \\
3x_1 + 2x_2 + 2x_3 + x_5^s = 23 &\ \ (w_2) \\
1x_1 + 2x_2 + 1x_3 + x_6^s = 17 &\ \ (w_3) \\
x_1, x_2, x_3, x_4^s, x_5^s, x_6^s \geq 0.&
\end{aligned}
$$

♠

The LP $\max\{\mathbf{cx} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq 0\}$ has an optimal solution $\mathbf{x}^*$ (the dual is $\min\{\mathbf{wb} : \mathbf{wA} \geq \mathbf{c}, \mathbf{w} \geq 0\}$).

- Since $\mathbf{x}^*$ is optimal there is no direction $\mathbf{d}$ such that $\mathbf{c}(\mathbf{x}^* + \lambda\mathbf{d}) > \mathbf{cx}^*$, $\mathbf{A}(\mathbf{x}^* + \lambda\mathbf{d}) \leq \mathbf{b}$, and $\mathbf{x}^* + \lambda\mathbf{d} \geq 0$ for $\lambda > 0$.

- Let $\mathbf{Gx} \leq \mathbf{g}$ be the binding inequalities in $\mathbf{Ax} \leq \mathbf{b}$ and $\mathbf{x} \geq 0$ for solution $\mathbf{x}^*$, that is, $\mathbf{Gx}^* = \mathbf{g}$.

  For our example,

$$
\mathbf{G}|\mathbf{g} = \left[\begin{array}{ccc|c} 3 & 2 & 2 & 23 \\ 1 & 2 & 1 & 17 \\ 0 & 0 & -1 & 0 \end{array}\right]
$$

- Based on the optimality of $\mathbf{x}^*$, there is no direction $\mathbf{d}$ at $\mathbf{x}^*$ such that $\mathbf{cd} > 0$ and $\mathbf{Gd} \leq 0$ (this includes $\mathbf{d} \leq 0$) (else we could improve the solution).

- From Farka's Lemma, if the system $\mathbf{cd} > 0$, $\mathbf{Gd} \leq 0$ does not have a solution, the system $\mathbf{wG} = \mathbf{c}$, $\mathbf{w} \geq 0$ must have a solution.

$$
\begin{aligned}
&3w_2 + 1w_3 = 18 \;\; (x_1)\\
&2w_2 + 2w_3 = 16 \;\; (x_2)\\
&2w_2 + 1w_3 - w_6^s = 10 \;\; (x_3)\\
&w_2, w_3, w_6^s, \geq 0.
\end{aligned}
$$

D: Min	z_D	w_1	w_2	w_3	w_4^s	w_5^s	w_6^s	rhs
z_D	1	-1	0	0	-3	-7	0	166
w_2	0	1/2	1	0	-1/2	1/4	0	5
w_3	0	1/2	0	1	1/2	-3/4	0	3
w_6^s	0	1/2	0	0	-1/2	-1/4	1	3

Challenge 1: Solve the following LP (as represented in the tableau), using the given tableau as a starting point. Provide the details of the algorithm to do so, and make it valid for both maximization and minimization problems.

D : min	z_D	w_1	w_2	w_3	w_4^s	w_5^s	w_6^s	rhs
z_D	1	-21	-23	-17	0	0	0	0
w_4^s	0	-2	-3	-1	1	0	0	-18
w_5^s	0	-2	-2	-2	0	1	0	-16
w_6^s	0	-1	-2	-1	0	0	1	-10

Challenge 2: Given the following optimal tableau to our production LP, we can buy 12 units of resource 2 for \$4 a unit. Should we, please provide the analysis needed to make this decision.

P : max	z_P	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	rhs
z_P	1	0	0	3	0	5	3	166
x_4^s	0	0	0	-1/2	1	-1/2	-1/2	1
x_1	0	1	0	1/2	0	1/2	-1/2	3
x_2	0	0	1	1/4	0	-1/4	3/4	7

Buying more of a resource: In the example problem,we currently have 23 units of the second resource, $b_2 = 23$. We want to know the range of b_2-values for which the current basis remains feasible. The formula $\mathbf{B}^{-1}\mathbf{b}$ shows us the impact of changing b_2 (which only changes the *rhs* column of the tableau). So, we set $\mathbf{B}^{-1}\mathbf{b} \geq 0$ replacing 23 with unknown b_2, and solve.

$$\begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & 1/2 & -1/2 \\ 0 & -1/4 & 3/4 \end{bmatrix} \begin{bmatrix} 21 \\ b_2 \\ 17 \end{bmatrix} \geq 0 \Rightarrow 17 \leq b_2 \leq 25.$$

P : max	z_P	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	rhs
z_P	1	0	0	3	0	5	3	166
x_4^s	0	0	0	-1/2	1	-1/2	-1/2	1
x_1	0	1	0	1/2	0	1/2	-1/2	3
x_2	0	0	1	1/4	0	-1/4	3/4	7

If $17 \leq b_2 \leq 25$, then the current basis remains feasible (and optimal). The *shadow price* for this resource is 5, thus 10 is the break-even price for the two additional units of resource 2. If 4 units of resource 2 were for sale, what is the break-even price?

If we add these additional resources, and recalculate the tableau, we get the following:

z	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	rhs
1	0	0	3	0	5	3	186
0	0	0	-1/2	1	-1/2	-1/2	-1
0	1	0	1/2	0	1/2	-1/2	5
0	0	1	1/4	0	-1/4	3/4	6

This tableau looks optimal (see row zero), but the basis is infeasible. We can find a new basis that is feasible and still looks optimal using the *Dual Simplex Method*.

Here we find the current basic variable with the smallest negative *rhs* coefficient, in this case there is only one negative coefficient, and that is for x_4^s. This is the leaving variable.

To find the entering variable, use the ration test and pivot. Note that in this case either x_3 or x_6^s can enter (they tie in the ratio test), and either route leads to an optimal solution (there are multiple optimal solutions here). If we enter x_3 into the basis we get the following tableau:

z	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	rhs
1	0	0	0	6	2	0	180
0	0	0	1	-2	1	1	2
0	1	0	0	1	0	-1	4
0	0	1	0	1/2	-1/2	1/2	11/2

Thus, the break-even price for the 4 units of resource 2 is 14.

Adding a new constraint: Given an optimal basis (i.e., tableau), what does adding a new constraint do? Consider a new resource having the following constraint:

$$3/2x_1 + 3/2x_2 + 3/2x_3 \leq 14.$$

We can enter this into the current optimal tableau:

z	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	x_7^s	rhs
1	0	0	3	0	5	3	0	166
0	0	0	-1/2	1	-1/2	-1/2	0	1
0	1	0	1/2	0	1/2	-1/2	0	3
0	0	1	1/4	0	-1/4	3/4	0	7
0	3/2	3/2	3/2	0	0	0	1	14

This tableau no longer looks like one having a basic solution, so using elementary row operations, we get the following:

z	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	x_7^s	rhs
1	0	0	3	0	5	3	0	166
0	0	0	-1/2	1	-1/2	-1/2	0	1
0	1	0	1/2	0	1/2	-1/2	0	3
0	0	1	1/4	0	-1/4	3/4	0	7
0	0	0	3/8	0	-3/8	-3/8	1	-1

This tableau is no longer feasible, so using dual simplex we obtain the following (x_7^s leaves the basis and x_6^s enters).

z	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	x_7^s	rhs
1	0	0	6	0	2	0	8	158
0	0	0	-1	1	0	0	-4/3	7/3
0	1	0	0	0	1	0	-4/3	13/3
0	0	1	1	0	-1	0	2	5
0	0	0	-1	0	1	1	-8/3	8/3

What if the constraint was

$$3/2x_1 + 3/2x_2 + 3/2x_3 = 14.$$

What if the constraint was

$$3/2x_1 + 3/2x_2 + 3/2x_3 = 18.$$

Using x_7 as an artificial variable, we get the following tableau (an additional row labeled z_a is added for the phase 1 problem).

	z	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	x_7^a	RHS
z	1	0	0	3	0	5	3	0	166
z_a	0	0	0	0	0	0	0	-1	0
x_4	0	0	0	-1/2	1	-1/2	-1/2	0	1
x_1	0	1	0	1/2	0	1/2	-1/2	0	3
x_2	0	0	1	1/4	0	-1/4	3/4	0	7
x_7	0	0	0	3/8	0	-3/8	-3/8	1	3

Adjusting this tableau to represent the initial, artificial, basis yields:

	z	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	x_7^a	RHS
z	1	0	0	3	0	5	3	0	166
z_a	0	0	0	3/8	0	-3/8	-3/8	0	3
x_4	0	0	0	-1/2	1	-1/2	-1/2	0	1
x_1	0	1	0	1/2	0	1/2	-1/2	0	3
x_2	0	0	1	1/4	0	-1/4	3/4	0	7
x_7	0	0	0	3/8	0	-3/8	-3/8	1	3

Since the phase 1 problem is a minimization, we enter x_3 into the basis, and remove x_1, this yields an optimal solution to the phase 1 problem having the artificial variable x_7 in the basis, thus with the new constraint, the LP is infeasible.

Adding a new variable: Consider the following tableau, to which a new column has been added (corresponding to x_7), given this column, please find the original data for this variable, i.e., the values of c_7 and $\mathbf{a}_7$.

z	x_1	x_2	x_3	x_4^s	x_5^s	x_6^s	x_7	RHS
1	0	0	3	0	5	3	-1	166
0	0	0	-1/2	1	-1/2	-1/2	0	1
0	1	0	1/2	0	1/2	-1/2	0	3
0	0	1	1/4	0	-1/4	3/4	1/2	7

4.2.6. Theory Applications

Cutting Stock - Column Generation

Consider the Cutting Stock Problem, which we use to illustrate column generation:

Given a stock board of length q and demand d_i for boards of length l_i (where $l_i \leq q$), you must cut the stock boards to satisfy this demand, while minimizing waste, i.e., the number of stock boards required to satisfy the demand.

Problem parameters:

P set of cutting pattern indexes, $\{1, 2, \cdots, n\}$.

L set of board length indexes, $\{1, 2, \cdots, m\}$.

d_i demand for boards of length l_i, $i = 1, \cdots, m$.

a_{ij} number of boards of length l_i, obtained when one stock board is cut using pattern j, $i \in L$, $j \in P$.

Decision variable:

x_j number of stock boards to cut using pattern $j \in P$.

$$\begin{aligned} \text{Min } \; & z = \sum_{j \in P} x_j \\ \text{s.t. } \; & \sum_{j \in P} a_{ij} x_j \geq d_i, \; \forall i \in L \\ & x_j \geq 0, \; \forall j \in P. \end{aligned}$$

It can be difficult to enumerate all possible cutting patterns $\mathbf{a}_j$ (this set can be quite large).

Instead, solve a restricted problem (as follows) where P_R is a subset of P that provides a feasible solution:

$$
\begin{aligned}
\text{Min } & z = \sum_{j \in P_R} x_j \\
\text{s.t. } & \sum_{j \in P_R} a_{ij} x_j \geq d_i, \ \forall i \in L \\
& x_j \geq 0, \ \forall j \in P_R.
\end{aligned}
$$

The optimal solution to the restricted problem is a feasible solution to the full problem. We want to find a new cutting pattern that will allow us to improve the restricted problem solution. Recall that the optimality condition for a minimization is $\mathbf{c_B B^{-1} a_i} - \mathbf{c_i} \leq 0$, thus to improve the restricted problem we want a column defined by $\mathbf{a_i}$ such that $\mathbf{c_B B^{-1} a_i} - \mathbf{c_i} > 0$. For this problem $c_i = 1, i \in P$.

To find this vector $\mathbf{a_i}$ (a column in the simplex tableau and a cutting pattern), we use the optimal solution to the restricted primal, which defines $\mathbf{c_B B^{-1}}$, we can solve the following integer program:

$$
\begin{aligned}
\text{Max } & \sum_{i \in L} \mathbf{c_B B}^{-1} a_i \\
\text{s.t. } & \sum_{i \in L} l_i a_i \leq q, \\
& a_i \in \mathscr{Z}^{\geq 0}, \ \forall i \in L,
\end{aligned}
$$

where the a_i's are the decision variables. This produces a new cutting pattern, which is then added to the restricted problem. This process continues until the sub-problem provides an optimal solution of zero.

Revenue Managemen - Shadow Prices

Consider an airline with a hub-and-spoke route structure. We define a flight as one take-off and landing of an aircraft at a particular time, flight are usually given flight numbers. Ticket prices are based on the itinerary, where an itinerary is a specific flight or set of (connecting) flights and a booking class (reflated to rules for the ticket, e.g., refundability). The following figure illustrates seven possible combinations that can be used to build itineraries using connections through airport B, the hub (A-B, B-C, B-D, B-E, A-B-C, A-B-D, A-B-E).

The airline forecasts demand for all important itinerary. Here are the problem parameters.

Problem Parameters:

F set of all flights (one take-off and landing of an aircraft, at a specific time).

c_f capacity of flight $f, \forall f \in F$.

I set of all itineraries (set of flights that customer uses) and booking class.

I_f set of all itineraries on flight f, $\forall f \in F$.

d_i demand for itinerary i, $\forall i \in I$.

f_i fare for itinerary i, $\forall i \in I$.

Decision Variables:

x_i # of passengers accepted for itinerary i, $\forall i \in I$.

The following linear program maximize the revenue:

$$\begin{aligned} \text{Max} \quad & \sum_{i \in I} f_i x_i \\ \text{s.t.:} \quad & x_i \le d_i, \quad \forall i \in I \\ & \sum_{i \in I_f} x_i \le c_f, \quad \forall f \in F \\ & x_i \ge 0, \quad \forall i \in I \end{aligned}$$

4.3 Other material for Integer Linear Programming

Recall the problem on lemonade and lemon juice from Chapter **??**:

Problem. Say you are a vendor of lemonade and lemon juice. Each unit of lemonade requires 1 lemon and 2 litres of water. Each unit of lemon juice requires 3 lemons and 1 litre of water. Each unit of lemonade gives a profit of \$3. Each unit of lemon juice gives a profit of \$2. You have 6 lemons and 4 litres of water available. How many units of lemonade and lemon juice should you make to maximize profit?

Letting x denote the number of units of lemonade to be made and letting y denote the number of units of lemon juice to be made, the problem could be formulated as the following linear programming problem:

$$\begin{array}{rrcrcl} \max & 3x & + & 2y & & \\ \text{s.t.} & x & + & 3y & \le & 6 \\ & 2x & + & y & \le & 4 \\ & x & & & \ge & 0 \\ & & & y & \ge & 0. \end{array}$$

The problem has a unique optimal solution at $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1.2 \\ 1.6 \end{bmatrix}$ for a profit of 6.8. But this solution requires us to make fractional units of lemonade and lemon juice. What if we require the number of units to be integers? In other words, we want to solve

$$\begin{array}{rrcrcl} \max & 3x & + & 2y & & \\ \text{s.t.} & x & + & 3y & \leq & 6 \\ & 2x & + & y & \leq & 4 \\ & x & & & \geq & 0 \\ & & & y & \geq & 0 \\ & x & , & y & \in & \mathbb{Z}. \end{array}$$

This problem is no longer a linear programming problem. But rather, it is an integer linear programming problem.

A **mixed-integer linear programming problem** is a problem of minimizing or maximizing a linear function subject to finitely many linear constraints such that the number of variables are finite and at least one of which is required to take on integer values.

If all the variables are required to take on integer values, the problem is called a **pure integer linear programming problem** or simply an **integer linear programming problem**. Normally, we assume the problem data to be rational numbers to rule out some pathological cases.

Mixed-integer linear programming problems are in general difficult to solve yet they are too important to ignore because they have a wide range of applications (e.g. transportation planning, crew scheduling, circuit design, resource management etc.) Many solution methods for these problems have been devised and some of them first solve the **linear programming relaxation** of the original problem, which is the problem obtained from the original problem by dropping all the integer requirements on the variables.

Example 4.6

Let (MP) denote the following mixed-integer linear programming problem:

$$\begin{array}{rrcrcrcl} \min & x_1 & & & + & x_3 & & \\ \text{s.t.} & -x_1 & + & x_2 & + & x_3 & \geq & 1 \\ & -x_1 & - & x_2 & + & 2x_3 & \geq & 0 \\ & -x_1 & + & 5x_2 & - & x_3 & = & 3 \\ & x_1 & , & x_2 & , & x_3 & \geq & 0 \\ & & & & & x_3 & \in & \mathbb{Z}. \end{array}$$

The linear programming relaxation of (MP) is:

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

Let (P1) denote the linear programming relaxation of (MP). Observe that the optimal value of (P1) is a lower bound for the optimal value of (MP) since the feasible region of (P1) contains all the feasible solutions to (MP), thus making it possible to find a feasible solution to (P1) with objective function value better than the optimal value of (MP). Hence, if an optimal solution to the linear programming relaxation happens to be a feasible solution to the original problem, then it is also an optimal solution to the original problem. Otherwise, there is an integer variable having a nonintegral value v. What we then do is to create two new subproblems as follows: one requiring the variable to be at most the greatest integer less than v, the other requiring the variable to be at least the smallest integer greater than v. This is the basic idea behind the **branch-and-bound method**. We now illustrate these ideas on (MP).

Solving the linear programming relaxation (P1), we find that $\mathbf{x}' = \begin{bmatrix} 0 \\ \frac{2}{3} \\ \frac{1}{3} \end{bmatrix}$ is an optimal solution to (P1). Note that $\mathbf{x}'$ is not a feasible solution to (MP) because x_3' is not an integer. We now create two subproblems (P2) and (P3) such that (P2) is obtained from (P1) by adding the constraint $x_3 \leq \lfloor x_3' \rfloor$ and (P3) is obtained from (P1) by adding the constraint $x_3 \geq \lceil x_3' \rceil$. (For a number a, $\lfloor a \rfloor$ denotes the greatest integer at most a and $\lceil a \rceil$ denotes the smallest integer at least a.) Hence, (P2) is the problem

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

and (P3) is the problem

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

Note that any feasible solution to (MP) must be a feasible solution to either (P2) or (P3). Using the help of a solver, one sees that (P2) is infeasible. The problem (P3) has an optimal solution at $\mathbf{x}^* = \begin{bmatrix} 0 \\ \frac{4}{5} \\ 1 \end{bmatrix}$, which is also feasible to (MP). Hence, $\mathbf{x}^* = \begin{bmatrix} 0 \\ \frac{4}{5} \\ 1 \end{bmatrix}$ is an optimal solution to (MP).

We now give a description of the method for a general mixed-integer linear programming problem (MIP). Suppose that (MIP) is a minimization problem and has n variables $x_1, \ldots, x_n$. Let $\mathscr{I} \subseteq \{1, \ldots, n\}$ denote the set of indices i such that x_i is required to be an integer in (MIP).

Branch-and-bound method

Input: The problem (MIP).

Steps:

1. Set `bestbound` $:= \infty$, $\mathbf{x}^*_{\text{best}} :=$ `N/A`, `activeproblems` $:= \{(LP)\}$ where (LP) denotes the linear programming relaxation of (MIP).

2. If there is no problem in `activeproblems`, then stop; if $\mathbf{x}^*_{\text{best}} \neq$ `N/A`, then $\mathbf{x}^*_{\text{best}}$ is an optimal solution; otherwise, (MIP) has no optimal solution.

3. Select a problem P from `activeproblems` and remove it from `activeproblems`.

4. Solve P.

- If P is unbounded, then stop and conclude that (MIP) does not have an optimal solution.

- If P is infeasible, go to step 2.

- If P has an optimal solution $\mathbf{x}^*$, then let z denote the objective function value of $\mathbf{x}^*$.

5. If $z \geq$ `bestbound`, go to step 2.

6. If x_i^* is not an integer for some $i \in \mathscr{I}$, then create two subproblems P_1 and P_2 such that P_1 is the problem obtained from P by adding the constraint $x_i \leq \lfloor x_i^* \rfloor$ and P_2 is the problem obtained from P by adding the constraint $x_i \geq \lceil x_i^* \rceil$. Add the problems P_1 and P_2 to `activeproblems` and go to step 2.

7. Set $\mathbf{x}^*_{\text{best}} = \mathbf{x}^*$, `bestbound` $= z$ and go to step 2.

Remarks.

- Throughout the algorithm, `activeproblems` is a set of subproblems remained to be solved. Note that for each problem P in `activeproblems`, P is a linear programming problem and that any feasible solution to P satisfying the integrality requirements is a feasible solution to (MIP).

- x^*_{best} is the feasible solution to (MIP) that has the best objective function value found so far and `bestbound` is its objective function value. It is often called an **incumbent**.

- In practice, how a problem from `activeproblems` is selected in step 3 has an impact on the overall performance. However, there is no general rule for selection that guarantees good performance all the time.

- In step 5, the problem P is discarded since it cannot contain any feasible solution to (MIP) having a better objective function value than x^*_{best}.

- If step 7 is reached, then x^* is a feasible solution to (MIP) having objective function value better than `bestbound`. So it becomes the current best solution.

- It is possible for the algorithm to never terminate. Below is an example for which the algorithm will never stop:

$$
\begin{array}{rccccccl}
\min & x_1 \\
\text{s.t.} & x_1 & + & 2x_2 & - & 2x_3 & = & 1 \\
& x_1 & , & x_2 & , & x_3 & \geq & 0 \\
& x_1 & , & x_2 & , & x_3 & \in & \mathbb{Z}.
\end{array}
$$

However, it is easy to see that $\mathbf{x}^* = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ is an optimal solution because there is no feasible solution with $x_1 = 0$.

One way to keep track of the progress of the computations is to set up a progress chart with the following headings:

Iter	**solved**	**status**	**branching**	`activeproblems`	$\mathbf{x}^*_{\text{best}}$	`bestbound`

In a given iteration, the entry in the **solved** column denotes the subproblem that has been solved with the result in the **status** column. The **branching** column indicates the subproblems created from the solved subproblem with an optimal solution not feasible to (MIP). The entries in the remaining columns contain the latest information in the given iteration. For the example (MP) above, the chart could look like the following:

Iter	**solved**	**status**	**branching**	`activeproblems`	$\mathbf{x}^*_{\text{best}}$	`bestbound`
1	(P1)	optimal $\mathbf{x}^* = \begin{bmatrix} 0 \\ \frac{2}{3} \\ \frac{1}{3} \end{bmatrix}$	(P2): $x_3 \leq 0$, (P3): $x_3 \geq 1$	(P2), (P3)	N/A	∞
2	(P2)	infeasible	—	(P3)	N/A	∞
3	(P3)	optimal $\mathbf{x}^* = \begin{bmatrix} 0 \\ \frac{4}{5} \\ 1 \end{bmatrix}$	—	—	$\begin{bmatrix} 0 \\ \frac{4}{5} \\ 1 \end{bmatrix}$	1

Exercises

1. Suppose that (MP) in Example **??** above has x_2 required to be an integer as well. Continue with the computations and determine an optimal solution to the modified problem.

2. With the help of a solver, determine the optimal value of

$$\begin{array}{rrcrcl} \max & 3x & + & 2y & & \\ \text{s.t.} & x & + & 3y & \leq & 6 \\ & 2x & + & y & \leq & 4 \\ & x & , & y & \geq & 0 \\ & x & , & y & \in & \mathbb{Z}. \end{array}$$

3. Let $\mathbf{A} \in \mathbb{Q}^{m\times n}$ and $\mathbf{b} \in \mathbb{Q}^m$. Let S denote the system

$$\begin{aligned} \mathbf{Ax} &\geq \mathbf{b} \\ \mathbf{x} &\in \mathbb{Z}^n \end{aligned}$$

   a. Suppose that $\mathbf{d} \in \mathbb{Q}^m$ satisfies $\mathbf{d} \geq 0$ and $\mathbf{d}^\top \mathbf{A} \in \mathbb{Z}^n$. Prove that every $\mathbf{x}$ satisfying S also satisfies $\mathbf{d}^T \mathbf{Ax} \geq \lceil \mathbf{d}^\top \mathbf{b} \rceil$. (This inequality is known as a **Chvátal-Gomory cutting plane.**)

   b. Suppose that $\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 5 & 3 \\ 7 & 6 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} 2 \\ 1 \\ 8 \end{bmatrix}$. Show that every $\mathbf{x}$ satisfying S also satisfies $x_1 + x_2 \geq 2$.

Solutions

1. An optimal solution to the modified problem is given by $x^* = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$.

2. An optimal solution is $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$. Thus, the optimal value is 6.

3. a. Since $\mathbf{d} \geq 0$ and $\mathbf{Ax} \geq \mathbf{b}$, we have $\mathbf{d}^T \mathbf{Ax} \geq \mathbf{d}^\top \mathbf{b}$. If $\mathbf{d}^\top \mathbf{b}$ is an integer, the result follows immediately. Otherwise, note that $\mathbf{d}^\top \mathbf{A} \in \mathbb{Z}^n$ and $\mathbf{x} \in \mathbb{Z}^n$ imply that $\mathbf{d}^T \mathbf{Ax}$ is an integer. Thus, $\mathbf{d}^T \mathbf{Ax}$ must be greater than or equal to the least integer greater than $\mathbf{d}^\top \mathbf{b}$.

   b. Take $\mathbf{d} = \begin{bmatrix} \frac{1}{9} \\ 0 \\ \frac{1}{9} \end{bmatrix}$ and apply the result in the previous part.