AARHUS UNIVERSITET

Optimization/Linear Programming 1

- Geometric method

- Simplex method

Henrik Karstoft
@
Department of Engineering

Optimization/Linear Programming 1

Mathematical Optimization

$$\text{minimize } f_0(x)$$
$$\text{subject to } \; f_i(x) \leq b_i, \text{ where } i = 1, \dots, m$$

- $x = (x_1, \dots, x_n)$, Optimization variables, or decision variables
- $f_0\colon\ R^n \to R$, objective function
- $f_i(x)\ R^n \to R$, where $i = 1, \dots, m$: constraint functions
- x^* **optimal solution**, smallest value of f_0 among $x = (x_1, \dots, x_n)$, satisfying the constraint

Optimization/Linear Programming 1

Linear programming – examples ([Lay] p. 16)

EXAMPLE 1 The Shady-Lane grass seed company blends two types of seed mixtures, EverGreen and QuickGreen. Each bag of EverGreen contains 3 pounds of fescue seed, 1 pound of rye seed, and 1 pound of bluegrass. Each bag of QuickGreen contains 2 pounds of fescue, 2 pounds of rye, and 1 pound of bluegrass. The company has 1200 pounds of fescue seed, 800 pounds of rye seed, and 450 pounds of bluegrass available to put into its mixtures. The company makes a profit of $2 on each bag of EverGreen and $3 on each bag of QuickGreen that it produces. Set up the mathematical problem that determines the number of bags of each mixture that Shady-Lane should make in order to maximize its profit.

1 Bag	Fescue/p	Rye/p	Bluegrass/p
EverGreen	3	1	1
QuickGreen	2	2	1

$$
\begin{array}{lrl}
\text{Maximize} & 2x_1 + 3x_2 & \text{(profit function)} \\
\text{subject to} & 3x_1 + 2x_2 \le 1200 & \text{(fescue)} \\
 & x_1 + 2x_2 \le 800 & \text{(rye)} \\
 & x_1 + x_2 \le 450 & \text{(bluegrass)}
\end{array}
$$

and $x_1 \ge 0$, $x_2 \ge 0$.

Optimization/Linear Programming 1

Linear programming – examples ([CZ, 3e] p. 305, [CZ, 4e] p. 311)

Example 15.5 This example is adapted from [93]. Figure 15.1 shows an electric circuit that is designed to use a 30-V source to charge 10-V, 6-V, and 20-V batteries connected in parallel. Physical constraints limit the currents I_1, I_2, I_3, I_4, and I_5 to a maximum of 4 A, 3 A, 3 A, 2 A, and 2 A, respectively. In addition, the batteries must not be discharged; that is, the currents I_1, I_2, I_3, I_4, and I_5 must not be negative. We wish to find the values of the currents $I_1, \ldots, I_5$ such that the total power transferred to the batteries is maximized.

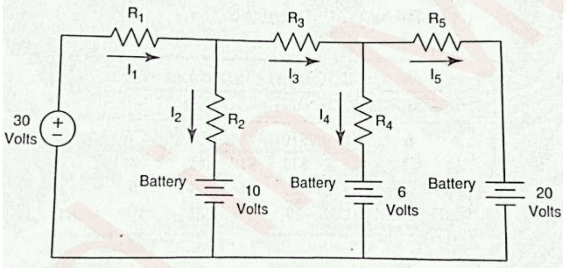


Figure 15.1 Battery charger circuit for Example 15.5.

$$
\begin{array}{ll}
\text{maximize} & 10I_2 + 6I_4 + 20I_5 \\
\text{subject to} & I_1 = I_2 + I_3 \\
& I_3 = I_4 + I_5 \\
& I_1 \leq 4 \\
& I_2 \leq 3 \\
& I_3 \leq 3 \\
& I_4 \leq 2 \\
& I_5 \leq 2 \\
& I_1, I_2, I_3, I_4, I_5 \geq 0.
\end{array}
$$

Optimization/Linear Programming 1

DEFINITION

Given $\mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$ in $\mathbb{R}^m$, $\mathbf{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}$ in $\mathbb{R}^n$, and an $m \times n$ matrix $A = [a_{ij}]$, the **canonical linear programming problem** is the following:

Find an n-tuple $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ in $\mathbb{R}^n$ to maximize

$$f(x_1, \dots, x_n) = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

subject to the constraints

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\le b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\le b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\le b_m \end{aligned}$$

and

$$x_j \ge 0 \quad \text{for } j = 1, \dots, n$$

This may be restated in vector-matrix notation as follows:

$$\text{Maximize } f(\mathbf{x}) = \mathbf{c}^T\mathbf{x} \quad (1)$$

$$\text{subject to the constraints } A\mathbf{x} \le \mathbf{b} \quad (2)$$

$$\text{and } \mathbf{x} \ge \mathbf{0} \quad (3)$$

where an inequality between two vectors applies to each of their coordinates.

Any vector $\mathbf{x}$ that satisfies (2) and (3) is called a **feasible solution**, and the set of all feasible solutions, denoted by $\mathcal{F}$, is called the **feasible set**. A vector $\bar{\mathbf{x}}$ in $\mathcal{F}$ is an **optimal solution** if $f(\bar{\mathbf{x}}) = \max_{\mathbf{x} \in \mathcal{F}} f(\mathbf{x})$.

Optimization/Linear Programming 1

Linear programming

$$\text{minimize } c^T x$$
$$\text{subject to } {a_i}^T x \leq b_i \; i = 1, \ldots, , m$$

- *no* analytic solution
- reliable and efficient algorithms and software
- computational time proportional to $n^2 m$, where $A \in R^{m \times n}$, in some case less than this
- mature technology

Optimization/Linear Programming 1

THEOREM 6 If the feasible set $\mathscr{F}$ is nonempty and if the objective function is bounded above on $\mathscr{F}$, then the canonical linear programming problem has at least one optimal solution. Furthermore, at least one of the optimal solutions is an extreme point of $\mathscr{F}$.[1]

Optimization/Linear Programming 1

EXAMPLE 5 Maximize $f(x_1, x_2) = 2x_1 + 3x_2$

$$\begin{aligned} \text{subject to} \quad x_1 \quad\;\; &\le 30 \\ x_2 &\le 20 \\ x_1 + 2x_2 &\le 54 \\ \text{and } x_1 \ge 0,\ x_2 \ge 0. & \end{aligned}$$

Since we only have 2 variables, the number of corners are limited: K(n+m, n)

(x_1, x_2)	$2x_1 + 3x_2$
(0, 0)	0
(30, 0)	60
(30, 12)	96 ←
(14, 20)	88
(0, 20)	60

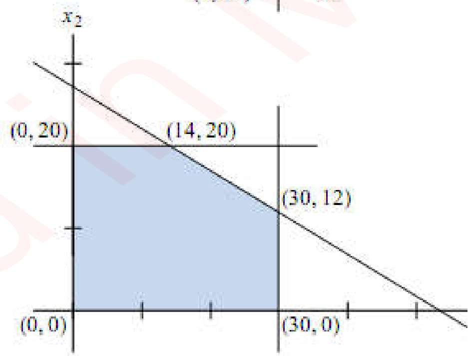


FIGURE 1

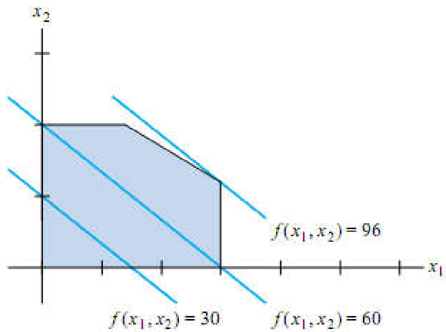


FIGURE 2

Optimization/Linear Programming 1

DEFINITION A **slack variable** is a nonnegative variable that is added to the smaller side of an inequality to convert it to an equality.

1. *A solution x to $[A, I]x = b$ is called basic if no more than m entries in x are nonzero (m is the number of rows in A).*

2. *A basic solution is a feasible, if all entries are nonnegative.*

EXAMPLE 3 Find a basic feasible solution for the system

$$\begin{aligned} 2x_1 + 3x_2 + 4x_3 &\le 60 \\ 3x_1 + x_2 + 5x_3 &\le 46 \\ x_1 + 2x_2 + x_3 &\le 50 \end{aligned}$$

Solution Add slack variables to obtain a system of three *equations*:

$$\begin{aligned} 2x_1 + 3x_2 + 4x_3 + x_4 \qquad\qquad &= 60 \\ 3x_1 + x_2 + 5x_3 \qquad + x_5 \qquad &= 46 \\ x_1 + 2x_2 + x_3 \qquad\qquad + x_6 &= 50 \end{aligned} \tag{1}$$

$$x_1 = x_2 = x_3 = 0, \quad x_4 = 60, \quad x_5 = 46, \quad \text{and} \quad x_6 = 50$$

A basic solution corresponds to corners.

Optimization/Linear Programming 1

EXAMPLE 4 Determine which row to use as a pivot in order to bring x_2 into the solution in Example 3.

Solution Compute the ratios b_i/a_{i2}:

$$\frac{b_1}{a_{12}} = \frac{60}{3} = 20, \qquad \frac{b_2}{a_{22}} = 46, \qquad \text{and} \qquad \frac{b_3}{a_{32}} = \frac{50}{2} = 25$$

b cannot be negative because otherwise they are not in the feasible set. Thus, we determine the ratios bi/ai2. By taking the smallest ratio, we ensure that b is always positive.

EXAMPLE 5 Maximize $25x_1 + 33x_2 + 18x_3$

$$\begin{aligned} \text{subject to} \qquad 2x_1 + 3x_2 + 4x_3 &\leq 60 \\ 3x_1 + x_2 + 5x_3 &\leq 46 \\ x_1 + 2x_2 + x_3 &\leq 50 \end{aligned}$$

and $x_j \geq 0$ for $j = 1, \ldots, 3$.

The value of M corresponds to the value of the objective function given the basic solution.

Optimization/Linear Programming 1

THE SIMPLEX ALGORITHM FOR A CANONICAL LINEAR PROGRAMMING PROBLEM

1. Change the inequality constraints into equalities by adding slack variables. Let M be a variable equal to the objective function, and below the constraint equations write an equation of the form

$$\text{(objective function)} - M = 0$$

2. Set up the initial simplex tableau. The slack variables (and M) provide the initial basic feasible solution.
3. Check the bottom row of the tableau for optimality. If all the entries to the left of the vertical line are nonnegative, then the solution is optimal. If some are negative, then choose the variable x_k for which the entry in the bottom row is as negative as possible.[3]
4. Bring the variable x_k into the solution. Do this by pivoting on the positive entry a_{pk} for which the nonnegative ratio b_i/a_{ik} is the smallest. The new basic feasible solution includes an increased value for M.
5. Repeat the process, beginning at step 3, until all the entries in the bottom row are nonnegative.

Optimization/Linear Programming 1

EXAMPLE 6 A health food store sells two different mixtures of nuts. A box of the first mixture contains 1 pound of cashews and 1 pound of peanuts. A box of the second mixture contains 1 pound of filberts and 2 pounds of peanuts. The store has available 30 pounds of cashews, 20 pounds of filberts, and 54 pounds of peanuts. Suppose the profit on each box of the first mixture is \$2 and on each box of the second mixture is \$3. If the store can sell all of the boxes it mixes, how many boxes of each mixture should be made in order to maximize the profit?

$$
\begin{array}{lrl}
\text{Maximize} & 2x_1 + 3x_2 & \\
\text{subject to} & x_1 \qquad\quad \leq 30 & \text{(cashews)} \\
 & x_2 \leq 20 & \text{(filberts)} \\
 & x_1 + 2x_2 \leq 54 & \text{(peanuts)} \\
\end{array}
$$

and $x_1 \geq 0, x_2 \geq 0.$

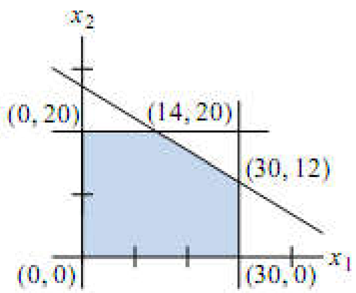


FIGURE 1

Optimization/Linear Programming 1

The solution for the dual problem corresponds to the solution of the column player in the matrix games.

EXAMPLE 8 Minimize $5x_1 + 3x_2$

$$\begin{aligned} \text{subject to} \quad 4x_1 + x_2 &\geq 12 \\ x_1 + 2x_2 &\geq 10 \\ x_1 + 4x_2 &\geq 16 \end{aligned}$$

and $x_1 \geq 0,\ x_2 \geq 0.$

$$\begin{aligned} -4x_1 - x_2 + x_3 \phantom{{}+x_4+x_5+M} &= -12 \\ -x_1 - 2x_2 \phantom{{}+x_3} + x_4 \phantom{{}+x_5+M} &= -10 \\ -x_1 - 4x_2 \phantom{{}+x_3+x_4} + x_5 \phantom{{}+M} &= -16 \\ 5x_1 + 3x_2 \phantom{{}+x_3+x_4+x_5} + M &= 0 \end{aligned}$$

$$\begin{array}{cccccc|c} x_1 & x_2 & x_3 & x_4 & x_5 & M & \\ -4 & -1 & 1 & 0 & 0 & 0 & -12 \\ -1 & -2 & 0 & 1 & 0 & 0 & -10 \\ -1 & -4 & 0 & 0 & 1 & 0 & -16 \\ \hline 5 & 3 & 0 & 0 & 0 & 1 & 0 \end{array}$$

However, the fastest method is to compute the usual ratios b_i/a_{ij} for all negative entries in rows 1 to 3 of columns 1 and 2. Choose as the pivot the entry with the *largest* ratio. That will make *all* the augmented entries change sign (because the pivot operation will *add* multiples of the pivot row to the other rows). In this example, the pivot should be a_{31}, and the new tableau is

Optimization/Linear Programming 1

Exampel Networkflow

We can model the max flow problem as a linear program too.

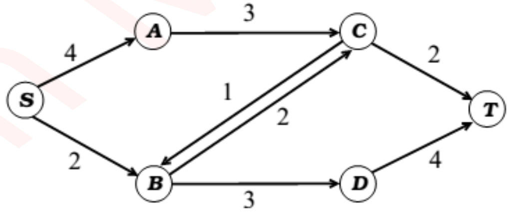
Variables: Set up one variable x_{uv} for each edge (u, v). Let's just represent the positive flow since it will be a little easier with fewer constraints.

Objective: Maximize $\sum_u x_{ut} - \sum_u x_{tu}$. (maximize the flow into t minus any flow out of t)

Constraints:

- For all edges (u, v), $0 \leq x_{uv} \leq c(u, v)$. (capacity constraints)
- For all $v \notin \{s, t\}$, $\sum_u x_{uv} = \sum_u x_{vu}$. (flow conservation)

For instance, consider the example from the network-flow lecture:



In this case, our LP is: maximize $x_{ct} + x_{dt}$ subject to the constraints:

$0 \leq x_{sa} \leq 4$, $0 \leq x_{ac} \leq 3$, etc.

$x_{sa} = x_{ac}$, $x_{sb} + x_{cb} = x_{bc} + x_{bd}$, $x_{ac} + x_{bc} = x_{cb} + x_{ct}$, $x_{bd} = x_{dt}$.

Optimization/Linear Programming 1

Matlab

- See: BB Demos/Lay CZ, Linear Programming and Optimization