**BDC6111: Introduction to Optimization** **Fall 2018, NUS**

# Lecture 1: Introduction

*Lecturer: Zhenyu Hu*

## 1.1 Basics of Optimization Problems

A generic optimization problem can be written as

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \mathcal{X}. \end{array}$$

Here $f : \mathcal{X} \to \mathbb{R}$ is called the *objective function*, its argument $x$ being the *decision variable(s).* The set $\mathcal{X}$ where $x$ can be taken from is called the *feasible set*, and its element the *feasible solution.*

For the most part of the lecture, we consider problems where $\mathcal{X} \subseteq \mathbb{R}^n$. In this case, $x = (x_1, ..., x_n)$.

Sometimes one may wish to deal with problems lying in infinite-dimensional space, e.g., $\mathcal{X}$ being a set of random variables. The underlying theory will not be explicitly dealt here (though many geometric insights are similar). Interested students are referred to David Luenberger's *Optimization by Vector Space Methods* for the classic book on this topic.

The set $\mathcal{X}$ in the generic optimization problem above is an abstract geometric concept and is quite often hard to handle analytically. More commonly, we are concerned with the following more explicit problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & h_i(x) = 0, i = 1, ..., m, \\ & x \in \mathbb{R}^n. \end{array}$$

Here, the equations $h_i(x) = 0$ are called the *constraints* (more specifically *equality constraints*). Clearly, with

$$\mathcal{X} = \{x \in \mathbb{R}^n | h_i(x) = 0, i = 1, ..., m\},$$

this is a special case of the generic problem we had at the beginning. The constraints here simply offer an algebraic characterization of the geometric object $\mathcal{X}$, and such characterization—as we shall see—is not necessarily unique.

If there exists $x^* \in \mathcal{X}$ such that $f(x^*) \leq f(x)$ for any $x \in \mathcal{X}$, then we call $x^*$ the *optimal solution* and $z = f(x^*)$ the *optimal value.* The existence of such $x^*$ is not guaranteed. In such case, we should replace min by inf and the optimal value is defined by $z = \inf\{f(x)|x \in \mathcal{X}\}$, which could be $-\infty$ (we then call the problem *unbounded*) or $+\infty$ (when $\mathcal{X} = \emptyset$).

### 1.1.1 More special forms

**Linear Programming Problem.** A general linear programming problem can be described as

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & a_i^T x \geq b_i, i \in M_1, \\ & a_i^T x \leq b_i, i \in M_2, \\ & a_i^T x = b_i, i \in M_3, \\ & x_j \geq 0, j \in N_1, \\ & x_j \leq 0, j \in N_2, \end{array} \tag{1.1}$$

where $M_1, M_2, M_3, N_1, N_2$ are some index sets and $c^T = (c_1, ..., c_n), a_i^T = (a_{i1}, ..., a_{in})$.

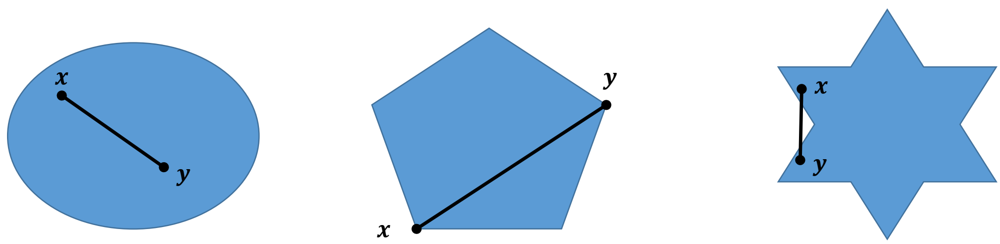
**Example 1.1** *Consider* $c^T = (2,5)$, $M_1 = \{1\}, M_2 = \{2\}, N_1 = \{1,2\}$ *and* $a_1^T = (1,1), b_1 = 6, a_2^T = (1,2), b_2 = 18$. *We then have*

$$\begin{array}{ll} \min & 2x_1 + 5x_2 \\ \text{s.t.} & x_1 + x_2 \geq 6, \\ & x_1 + 2x_2 \leq 18, \\ & x_1, x_2 \geq 0. \end{array}$$

**Convex Programming Problem.** A set $\mathcal{X}$ is *convex* if for any $x, y \in \mathcal{X}$, and any $0 \leq \lambda \leq 1$, we have

$$\lambda x + (1-\lambda) y \in \mathcal{X}.$$

The point $\lambda x + (1-\lambda)y$ is called the *convex combination* of $x$ and $y$. Some simple convex and nonconvex sets are illustrated in the figure below.




A function $f : \mathcal{X} \to \mathbb{R}$ is convex if $\mathcal{X}$ is a convex set and for any $x, y \in \mathcal{X}$ and $0 \leq \lambda \leq 1$, we have

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y).$$

A convex programming problem is then one of the form

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_i(x) \leq 0, i = 1, ..., l, \\ & a_i^T x = b_i, i = 1, ..., m, \end{array}$$

where $f(\cdot), g_1(\cdot), ..., g_l(\cdot)$ are convex functions. It can be shown that [exercise] the feasible set in this case:

$$\mathcal{X} = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} g_i(x) \leq 0, i = 1, ..., l \\ a_i^T x = b_i, i = 1, ..., m, \end{array} \right\}.$$

is a convex set.

Sometimes one may also refer to the abstract form

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \mathcal{X}, \end{array}$$

where $f$ is a convex function and $\mathcal{X}$ is a convex set as convex programming problem. However, in order to solve such problem, one still needs to find an algebraic description of the set using convex inequalities and linear equality constraints.

**Integer Programming Problem.** An integer programming problem is defined similarly to linear programming except the decision variables are constrained to take integer values. For instance:

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & a_i^T x = b_i, i = 1, ..., m \\ & x_j \geq 0, \\ & x \text{ integer}, \end{array}$$

is an integer programming. If only part of the variables are restricted to be integer values, then it is called a mixed integer programming problem. If furthermore, all the variables are constrained to be either 0 or 1, i.e., $x_j \in \{0, 1\}$, then the problem is called binary integer programming problem.

Integer programming problems are generally not convex, since convex combinations of integers may no longer be integer.

### 1.1.2 Comparison of optimization problems

Consider two generic optimization problems

$$\begin{array}{rl} z_1 = \min & f_1(x) \\ \text{s.t.} & x \in \mathcal{X} \end{array} \tag{1.2}$$

and

$$\begin{array}{rl} z_2 = \min & f_2(y) \\ \text{s.t.} & y \in \mathcal{Y} \end{array} \tag{1.3}$$

Note that $\mathcal{X}$ and $\mathcal{Y}$ are not necessarily the subsets of the same space.

We say problem (1.3) is a *relaxation* of problem (1.2) if for any $x \in \mathcal{X}$, there exists a $y \in \mathcal{Y}$ such that $f_2(y) = f_1(x)$. By definition, clearly we have $z_2 \leq z_1$ if (1.3) is a relaxation of (1.2). We also call (1.3) provides a *lower bound* to (1.2); and when it happens that $z_2 = z_1$, we say the lower bound (relaxation) is tight.

**Example 1.2 (Linear Relaxation)** *Consider problem* (1.2) *taking the following form*

$$\begin{array}{ll} \min & 2x_1 + 5x_2 \\ \text{s.t.} & x_1 + x_2 \geq 6, \\ & x_1 + 2x_2 \leq 18, \\ & x_1, x_2 \geq 0, \\ & x \text{ integer}. \end{array}$$

*Then its linear relaxation is defined as*

$$\begin{array}{ll} \min & 2x_1 + 5x_2 \\ \text{s.t.} & x_1 + x_2 \geq 6, \\ & x_1 + 2x_2 \leq 18, \\ & x_1, x_2 \geq 0. \end{array}$$

**Example 1.3 (Lagrangian Relaxation)** *Consider problem* (1.2) *taking the following form*

$$\begin{array}{ll} \min & f_1(x) \\ \text{s.t.} & g(x) = 0. \end{array}$$

*Then its Lagrangian relaxation is defined as*

$$\min \quad f_1(x) + \lambda g(x),$$

*where $\lambda \in \mathbb{R}$ is any fixed constant.*

**Example 1.4** *Consider problem* (1.2) *taking the following form*

$$\begin{array}{rl} \min & x_1 + x_2 \\ \text{s.t.} & x_1 + x_2 \le 1 \\ & x_1, x_2 \ge 0. \end{array}$$

*The following problem is a relaxation of it (it can be considered as a Lagrangian relaxation with $\lambda = 0$).*

$$\begin{array}{rl} \min & x_1 + x_2 \\ \text{s.t.} & x_1, x_2 \ge 0. \end{array}$$

*Note here that the relaxation is tight.*

We say problems (1.2) and (1.3) are *equivalent* if (1.2) is a relaxation of (1.3) and vice versa. In other words, given a feasible solution to one problem, we can always construct a feasible solution to the other, with the same cost. By definition $z_1^* = z_2^*$, and given an optimal solution to one problem we can construct an optimal solution to the other.

The second problem in Example 1.4—though being a tight relaxation of the first problem—they are NOT equivalent by the above definition. The distinction between equivalence and tight relaxation, however, is usually not important from a practical point of view since in both cases solving one problem we would get the solution to another.

**Example 1.5** *Consider problem* (1.2) *taking the following form*

$$\begin{array}{rl} \min & x_1 + x_2 \\ \text{s.t.} & x_1 + x_2 \le 1 \\ & x_1, x_2 \ge 0. \end{array}$$

*The following problem is equivalent to it.*

$$\begin{array}{rl} \min & x_1 + x_2 \\ \text{s.t.} & x_1 + x_2 \le 1 \\ & 0 \le x_1, x_2 \le 1. \end{array}$$

*Although the two problems admit different algebraic representations, from a geometric point of view, they are exactly the same problem.*

The following example shows a simple but non-trivial equivalence.

**Example 1.6** *Consider problem* (1.2) *taking the following form*

$$\min \quad |x|.$$

*The following problem is equivalent to it.*

$$\begin{array}{rl} \min & x_2 \\ \text{s.t.} & x_2 \ge x_1 \\ & x_2 \ge -x_1. \end{array}$$

*Given any feasible solution $x$ to the first problem, we can always let $x_2 = |x|, x_1 = x$ which yields the same cost. Given any feasible solution $(x_1, x_2)$ to the second problem, we can always let $x = x_2$ which again yields the same cost. The problem below, however, is not an equivalent problem.*

$$\begin{array}{ll} \min & x_2 \\ \text{s.t.} & x_2 \leq x_1 \\ & x_2 \leq -x_1. \end{array}$$

*For example, $(x_1, x_2) = (0, -1)$ is feasible here with cost $-1$; yet one cannot find a feasible solution $x$ such that $|x| = -1$. One can easily verify that it is not even a relaxation.*

The above example also shows that two equivalent problems do not need to be defined in the same space.

## 1.2 Linear Programming Problems

### 1.2.1 Standard form problem

A linear programming problem of the form

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b, \\ & x \geq 0, \end{array} \tag{1.4}$$

is said to be in *standard form*. Here, $A$ is a matrix of dimensions $m \times n$, i.e.,

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix},$$

and $b$ is a vector of dimension $m$, i.e., $b^T = (b_1, ..., b_m)$. We often use the following two alternative representations.

- Row representation

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & a_i^T x = b_i, i = 1, ..., m \\ & x \geq 0, \end{array}$$

  where $a_i^T = (a_{i1}, ..., a_{in})$.

- Column representation

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & \sum_{j=1}^n A_j x_j = b, \\ & x \geq 0, \end{array}$$

  where $A_j$ denotes the $j$-th column of $A$, that is,

$$A_j = \begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{bmatrix}.$$

The standard form problem is clearly a special case of the more general form introduced in (1.1). In fact, any linear programming problem can be equivalently transformed into standard form as well.

- *Elimination of inequality constraints:* Suppose one has an inequality constraint

$$a_i^T x \le b_i.$$

  The above constraint is equivalent to

$$a_i^T x + s_i = b_i, s_i \ge 0.$$

  The variable $s_i$ is called a *slack variable.* Similarly,

$$a_i^T x \ge b_i \iff a_i^T x - s_i = b_i, s_i \ge 0.$$

- *Elimination of free variables:* Suppose $x_j$ is an unrestricted variable. To transform it into equivalent standard form, we can replace $x_j$ by $x_j^+ - x_j^-$, where $x_j^+$ and $x_j^-$ are new variables with constraints $x_j^+, x_j^- \ge 0$.

- *Maximization to minimization:* Suppose one is interested in maximizing $c^T x$. Clearly, this is the same as minimizing $-c^T x$ (in terms of solution).

**Example 1.7** *The problem*

$$\begin{array}{rl} \min & 2x_1 + 4x_2 \\ \text{s.t.} & x_1 + x_2 \ge 3, \\ & 3x_1 + 2x_2 = 14, \\ & x_1 \ge 0, \end{array}$$

*is equivalent to the standard form problem*

$$\begin{array}{rl} \min & 2x_1 + 4x_2^+ - 4x_2^- \\ \text{s.t.} & x_1 + x_2^+ - x_2^- - x_3 = 3, \\ & 3x_1 + 2x_2^+ - 2x_2^- = 14, \\ & x_1, x_2^+, x_2^-, x_3 \ge 0. \end{array}$$

**Example 1.8** *The problem*

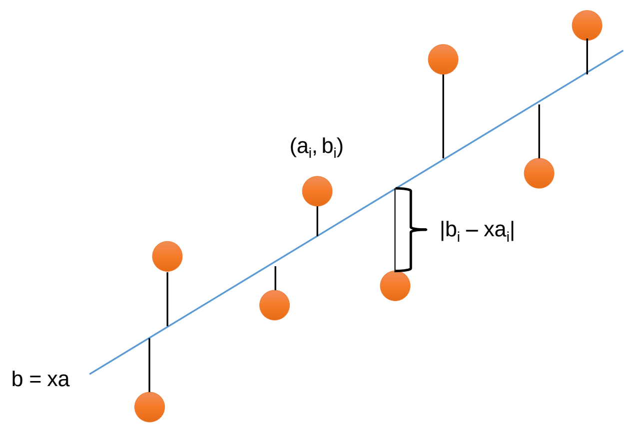
$$\begin{array}{rl} \max & c^T x \\ \text{s.t.} & Ax \le b, \\ & x \ge 0, \end{array}$$

*is equivalent to the standard form problem*

$$\begin{array}{rl} -\min & -c^T x \\ \text{s.t.} & [A \quad I] \begin{bmatrix} x \\ s \end{bmatrix} = b \\ & x, s \ge 0, \end{array}$$

### 1.2.2 Applications

**Data fitting problem.** We are given $m$ data points of the form $(a_i, b_i), i = 1, ..., m$, where $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$. We wish to find a linear model of the form $b = a^T x$, where $x$ are the parameters to be determined to fit those data points as good as possible. Given a fixed parameter vector $x$, the residual at the $i$-th data point is defined as $b_i - a_i^T x$. An illustration for the problem when $n = 1$ is given in the figure below.




- *Ordinary least squares ($l_2$):* This method seeks to minimize the sum of square of residuals, i.e.,

$$\min \sum_{i=1}^{m} (b_i - a_i^T x)^2.$$

  This problem is clearly nonlinear—it is in fact a quadratic programming problem.

- *Least absolute deviations ($l_1$):* This method seeks to minimize the sum of absolute value of residuals, i.e.,

$$\min \sum_{i=1}^{m} |b_i - a_i^T x|.$$

  While the problem appears to be a nonlinear problem, it can be "linearized" into a linear programming problem as we have seen in Example 1.6. In particular, it is equivalent to

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} y_i \\ \text{s.t.} \quad & b_i - a_i^T x \leq y_i, i = 1, ..., m \\ & -b_i + a_i^T x \leq y_i, i = 1, ..., m. \end{aligned}$$

- *Chebyshev approximation ($l_\infty$):* This method seeks to minimize the maximum of absolute residuals, i.e.,

$$\min \max_{1 \leq i \leq m} |b_i - a_i^T x|.$$

  Again, the problem can be casted as linear programming problem:

$$\begin{aligned} \min \quad & y \\ \text{s.t.} \quad & b_i - a_i^T x \leq y, i = 1, ..., m \\ & -b_i + a_i^T x \leq y, i = 1, ..., m. \end{aligned}$$

The above examples are special cases of either one of the following more general problems:

- *Norm approximation:* We seek to find an approximate solution $x$ such that $Ax \approx b$. This is achieved by
$$\min ||Ax - b||,$$
where $||\cdot||$ is a norm on $\mathbb{R}^m$.
- *Penalty function approximation:* Given the residuals $(b_1 - a_1^T x, ..., b_m - a_m^T x)$, one seeks to minimize some cost resulting from the these residuals
$$\min \sum_{i=1}^{m} \phi(b_i - a_i^T x),$$
where $\phi : \mathbb{R} \to \mathbb{R}$ is some penalty function that is assumed to be convex and nonnegative.

It would be helpful to compare $l_1$-norm and $l_2$-norm methods above from the following perspective:

- Difficulty in solving corresponding solution;
- Sensitivity to outliers;
- Solution uniqueness.

**Compressed sensing.** Consider a signal (e.g. sound, images):
$$x^T = (x_1, ..., x_n),$$
where $n$ is usually very large. The signal is not directly observable. Instead, one observes a measurement
$$b = Ax \in \mathbb{R}^m,$$
where $A$ is an $m \times n$ matrix and $m$ is small. The problem here is to recover the signal $x$ given the measurements $b$.

Clearly, the linear system of equations $Ax = b$ here is underdetermined, and the problem is in general unsolvable. Consider the following toy example.

**Example 1.9** *Suppose we have a two-dimensional unknown signal $x^T = (0, 1)$. The measurement matrix is*
$$A = [1 \quad 2].$$
*From the measurement, the only thing we can observe is $b = 2$. Clearly, any signal satisfying the equation $x_1 + 2x_2 = 2$ could be a solution.*

What if one has prior knowledge that the solution is *sparse*? Sparsity of the solution $x$ here means that most of its components are zero. In such case, one wish to solve the following problem
$$\begin{array}{ll} \min & |\{1 \le i \le n : x_i \neq 0\}| \\ \text{s.t.} & Ax = b, \end{array}$$
where $|\mathcal{S}|$ in the objective denotes the cardinality of the set $\mathcal{S}$. It is sometimes also called the $l_0$-"norm" of $x$[1].

The above problem, unfortunately, is computationally intractable (one can reformulate it as an integer programming problem). Based on idea that if $x$ has a small $l_0$-"norm", it should have a small $l_p$-norm as well, we can consider following two approximations:

[1]Note that, however, it is not a proper norm.

- *$l_2$-norm:*

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} x_i^2 \\ \text{s.t.} \quad & Ax = b. \end{aligned}$$

- *$l_1$-norm:*

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} |x_i| \\ \text{s.t.} \quad & Ax = b. \end{aligned}$$

It turns out that the $l_2$-norm problem can be a very bad approximation to the $l_0$-"norm" problem. On the other hand, the $l_1$-norm problem can be quite good, and under certain conditions can be exact as well (see [CRT06]).

**Piecewise linear convex objectives.** Let $c_1, ..., c_m$ be vectors in $\mathbb{R}^n$, let $d_1, ..., d_m$ be scalars, and consider the function $f : \mathbb{R}^n \to \mathbb{R}$ defined by

$$f(x) = \max_{i=1,\dots,m} (c_i^T x + d_i).$$

It can be shown that $f(x)$ is convex using the following more general result.

**Proposition 1.10** *Let $\mathcal{Y}$ be a non-empty set and let $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. Assume that $\mathcal{X}$ is a convex set and $g(\cdot, y)$ is convex for any $y \in \mathcal{Y}$. Then,*

$$\begin{aligned} f(x) = \max \quad & g(x, y) \\ \text{s.t.} \quad & y \in \mathcal{Y} \end{aligned}$$

*is a convex function on $\mathcal{X}$.*

*Proof:* Let $x, \tilde{x} \in \mathcal{X}$ and $\lambda \in [0, 1]$. Then,

$$\begin{aligned} f(\lambda x + (1-\lambda)\tilde{x}) = \max_{y \in \mathcal{Y}} \quad & g(\lambda x + (1-\lambda)\tilde{x}, y) \\ & \le \max_{y \in \mathcal{Y}} \{\lambda g(x, y) + (1-\lambda) g(\tilde{x}, y)\} \\ & \le \lambda \max_{y \in \mathcal{Y}} g(x, y) + (1-\lambda) \max_{y \in \mathcal{Y}} g(\tilde{x}, y) \\ & = \lambda f(x, y) + (1-\lambda) f(\tilde{x}, y). \end{aligned}$$

$\blacksquare$

By letting $g(x, i) = c_i^T x + d_i$, we know from above result that $f(x)$ is convex. A simple example of such function is $f(x) = |x| = \max\{x, -x\}$. Now consider the problem

$$\begin{aligned} \min \quad & \max_{i=1,\dots,m} (c_i^T x + d_i) \\ \text{s.t.} \quad & Ax \ge b. \end{aligned}$$

This problem can be equivalently formulated as

$$\begin{aligned} \min \quad & y \\ \text{s.t.} \quad & y \ge c_i^T x + d_i, i = 1, ..., m \\ & Ax \ge b. \end{aligned}$$

Similarly, the constraint

$$f(x) \leq h,$$

is equivalent to

$$c_i^T x + d_i \leq h, i = 1, ..., m.$$

# References


[BT97] Bertsimas, D. and J.N. Tsitsiklis, *Introduction to Linear Optimization*, Springer, 1997.

[CRT06] Candes, E. J., J. K. Romberg and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, 59(8), 2006, pp. 1207-1223.

BDC6111: Introduction to Optimization | Fall 2018, NUS

# Lecture 2: Geometry of LP

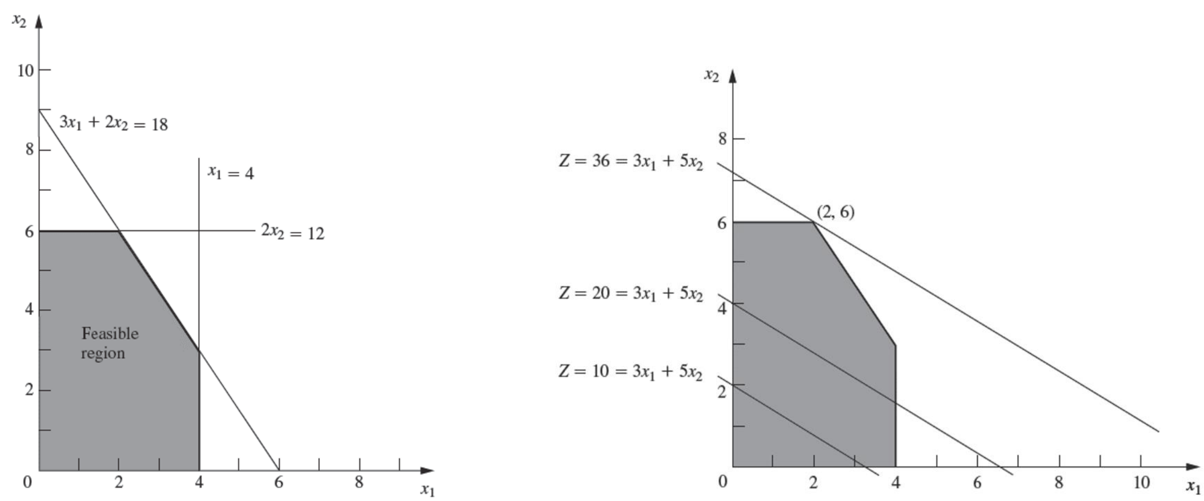
*Lecturer: Zhenyu Hu*

## 2.1 Graphical Approach

Consider the following two-dimensional linear programming problem

$$
\begin{aligned}
\max \quad & 3x_1 + 5x_2 \\
s.t. \quad & x_1 \leq 4, \\
& 2x_2 \leq 12, \\
& 3x_1 + 2x_2 \leq 18, \\
& x_1, x_2 \geq 0.
\end{aligned}
$$

Its feasible set is illustrated in Figure 2.1a.




(a) Feasible region

(b) Contour of the objective

Figure 2.1: Two dimensional example

The problem can be graphically solved by examining the contours of the objective $Z = 3x_1 + 5x_2$. From Figure 2.1b, one can see that the optimal solution is $(x_1^*, x_2^*) = (2, 6)$.

The central observation we made when solving this simple problem is that *the optimal solution must be some 'corner point' of the feasible set, not in the middle.* To generalize such observation to more general linear programs, we need to answer the following two central questions:

- What is a 'corner point' and how do we find it?
- Is it true that, in general, we only need to examine these 'corner points' to get the solution?

## 2.2 Linear Algebra Background

- We say a vector $\mathbf{y}$ is a *linear combination* of $\mathbf{x}^1, ..., \mathbf{x}^K$ if $\mathbf{y} = \sum_{k=1}^K a_k \mathbf{x}^k$ for some $a_k \in \mathbb{R}, k = 1, ..., K$.
- We say a collection of vectors $\mathbf{x}^1, ..., \mathbf{x}^K \in \mathbb{R}^n$ is *linearly dependent* if there exist real numbers $a_1, ..., a_K$, not all of them zero, such that $\sum_{k=1}^K a_k \mathbf{x}^k = \mathbf{0}$. On the other hand, if for any real numbers $a_1, ..., a_K$ such that $\sum_{k=1}^K a_k \mathbf{x}^k = \mathbf{0}$, we have $a_1 = ... = a_K = 0$, then $\mathbf{x}^1, ..., \mathbf{x}^K$ are *linearly independent*.
- Let $\mathbf{A}$ be a square matrix. If there exists a square matrix $\mathbf{B}$ such that
$$\mathbf{AB} = \mathbf{BA} = \mathbf{I},$$
then we say $\mathbf{A}$ is *invertible* or *nonsingular*. $\mathbf{B}$ is called the *inverse* of $\mathbf{A}$ and usually denoted as $\mathbf{A}^{-1}$.
- The determinant of a matrix $\mathbf{A}$ can be defined through the Laplace's formula:
$$\det(\mathbf{A}) = \sum_{j=1}^n (-1)^{i+j} a_{ij} \det(\mathbf{M}_{ij}), \quad \det(a_{ij}) = a_{ij}.$$
$\mathbf{M}_{ij}$ is called a minor of $\mathbf{A}$, which is a submatrix by removing the i-th row and j-th column of $\mathbf{A}$.

  **Remark 2.1** *If all elements in* $\mathbf{A}$ *are integers, is* $\det(\mathbf{A})$ *an integer?*

The following result states the equivalency among several fundamental concepts in linear algebra.

**Theorem 2.2** *Let* $\mathbf{A}$ *be an* $n \times n$ *matrix. Then, the following statements are equivalent:*

1. *The matrix* $\mathbf{A}$ *is invertible.*
2. *The determinant of* $\mathbf{A}$ *is nonzero, i.e.,* $|\mathbf{A}| \neq 0$ *or* $\det(\mathbf{A}) \neq 0$.
3. *The rows of* $\mathbf{A}$ *are linearly independent.*
4. *The columns of* $\mathbf{A}$ *are linearly independent.*
5. *For any n-dimensional vector* $\mathbf{b}$*, the linear system* $\mathbf{Ax} = \mathbf{b}$ *has a unique solution.*

**Example 2.3** *Consider the system of equations:*
$$\begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 40 \\ 30 \end{bmatrix}.$$

A nonempty subset $S$ of $\mathbb{R}^n$ is called a *subspace* of $\mathbb{R}^n$ if $a\mathbf{x} + b\mathbf{y} \in S$ for any $\mathbf{x}, \mathbf{y} \in S$ and any $a, b \in \mathbb{R}$. Note that any linear combinations of the elements of a subspace $S$ still lies in $S$. The *span* of $\mathbf{x}^1, ..., \mathbf{x}^K$ defined by the set of any linear combinations of $\mathbf{x}^1, ..., \mathbf{x}^K$, i.e.,
$$\{\mathbf{y} : \mathbf{y} = \sum_{k=1}^K a_k \mathbf{x}^k, \text{for any } \ a_k \in \mathbb{R}, k = 1, ..., K\}$$
is a subspace.

A *basis* of a subspace $S$ with $S \neq \{\mathbf{0}\}$ is a collection of vectors that are linearly independent and whose span is equal to $S$ and the number of vectors in the basis is called the *dimension* of the subspace. The set $\{\mathbf{0}\}$ is also a subspace, and we define its dimension to be 0.

**Example 2.4**

- *One-dimensional subspaces: lines through the origin;*
- *Two-dimensional subspace: planes through the origin.*

**Theorem 2.5** *Suppose that the span $S$ of the vectors $\mathbf{x}^1, ..., \mathbf{x}^K$ has dimension $m$. Then:*

1. *There exists a basis of $S$ consisting of $m$ of the vectors $\mathbf{x}^1, ..., \mathbf{x}^K$.*
2. *If $k \le m$ and $\mathbf{x}^1, ..., \mathbf{x}^k$ are linearly independent, we can form a basis of $S$ by starting with $\mathbf{x}^1, ..., \mathbf{x}^k$ and choosing $m - k$ of the vectors $\mathbf{x}^{k+1}, ..., \mathbf{x}^K$.*

**Example 2.6** *Suppose $S$ is defined by the span of vectors*

$$\mathbf{x}^1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \mathbf{x}^2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \mathbf{x}^3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

*The dimension of $S$ is 2. Any two vectors among $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3$ consist a basis for $S$.*

Let $\mathbf{A}$ be a matrix of dimensions $m \times n$. The *column space* of $\mathbf{A}$ is the subspace of $\mathbb{R}^m$ spanned by the columns of $\mathbf{A}$ and the *row space* is the subspace of $\mathbb{R}^n$ spanned by the rows of $\mathbf{A}$. The dimension of the column space is always equal to the dimension of the row space and this dimension is called the *rank* of $\mathbf{A}$, denoted as $\text{rank}(\mathbf{A})$. The set defined by $\{\mathbf{x} \in \mathbb{R}^n | \mathbf{A}\mathbf{x} = \mathbf{0}\}$ is a subspace of $\mathbb{R}^n$ and is called the *nullspace* of $\mathbf{A}$.

**Theorem 2.7 (rank-nullity theorem)** *The dimension of the nullspace is equal to $n - \text{rank}(\mathbf{A})$, where $n$ is the number of columns of $\mathbf{A}$.*

**Remark 2.8** *Suppose the $n \times n$ matrix $\mathbf{A}$ is invertible. What is the dimension of the nullspace of $\mathbf{A}$?*

## 2.3 A Geometric Perspective of the Solution

### Polyhedron

A *polyhedron* is a set that can be described in the form $\{\mathbf{x} \in \mathbb{R}^n | \mathbf{A}\mathbf{x} \ge \mathbf{b}\}$, where $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{b}$ is a vector in $\mathbb{R}^m$.

**Remark 2.9** *The feasible set of a standard form linear programming problem $\{\mathbf{x} \in \mathbb{R}^n | \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \ge \mathbf{0}\}$ is also a polyhedron.*

### Extreme Points and Vertices

Let $P$ be a polyhedron.

**Definition 2.10** *A vector* $\mathbf{x} \in P$ *is an* extreme point *of* $P$ *if we cannot find two vectors* $\mathbf{y}, \mathbf{z} \in P, \mathbf{y}, \mathbf{z} \neq \mathbf{x}$ *and* $\lambda \in [0, 1]$ *such that* $\mathbf{x} = \lambda\mathbf{y} + (1 - \lambda)\mathbf{z}$. *That is, an extreme point cannot be expressed as a convex combination of any other points in the polyhedron.*

**Remark 2.11** *The definition of extreme points does not refer to any representation of a polyhedron in terms of linear inequalities.*

An alternative concept is called the *vertex*.

**Definition 2.12** *A vector* $\mathbf{x} \in P$ *is a vertex of* $P$ *if there exists some* $\mathbf{c}$ *such that* $\mathbf{c}'\mathbf{x} < \mathbf{c}'\mathbf{y}$ *for all* $\mathbf{y} \in P$ *and* $\mathbf{y} \neq \mathbf{x}$.

In other words, $\mathbf{x} \in P$ is a vertex of $P$ if there exists $\mathbf{c}$ such that $\mathbf{x}$ is the unique optimal solution to the optimization problem

$$\begin{array}{ll} \min & \mathbf{c}'\mathbf{x} \\ \text{s.t.} & \mathbf{x} \in P. \end{array}$$

Geometrically, this implies $P$ is on one side of the hyperplane $\{\mathbf{y} | \mathbf{c}'\mathbf{y} = \mathbf{c}'\mathbf{x}\}$.

## 2.4 An Algebraic Perspective of the Solution

The geometric description given above is not easy to verify. Here, we try to describe the extreme point using the algebraic representation of the polyhedron.

### Basic Solution and Basic Feasible Solution

We start from a general description of polyhedron $P$ in $\mathbb{R}^n$:

$$\begin{aligned} \mathbf{a}_i'\mathbf{x} \geq b_i, &\quad i \in M_1 \\ \mathbf{a}_i'\mathbf{x} \leq b_i, &\quad i \in M_2, \\ \mathbf{a}_i'\mathbf{x} = b_i, &\quad i \in M_3. \end{aligned}$$

If a vector $\mathbf{x}^*$ satisfies $\mathbf{a}_i'\mathbf{x} = b_i$ for some $i$ in $M_1, M_2$ or $M_3$, then we say the corresponding constraint is *active* or *binding* at $\mathbf{x}^*$. If there are $n$ constraints active at $\mathbf{x}^*$, then $\mathbf{x}^*$ satisfies a linear system with $n$ linear equations and $n$ unknowns. If, in addition, these $n$ equations are linearly independent ($\{\mathbf{a}_i, i$ is a binding constraint$\}$ are linearly independent), then $\mathbf{x}^*$ is the unique solution of the system. This leads to the following definitions of *basic solution* and *basic feasible solution*.

**Definition 2.13** *The vector* $\mathbf{x}^* \in \mathbb{R}^n$ *is said to be a* basic solution *if:*

1. *All equality constraints are active;*

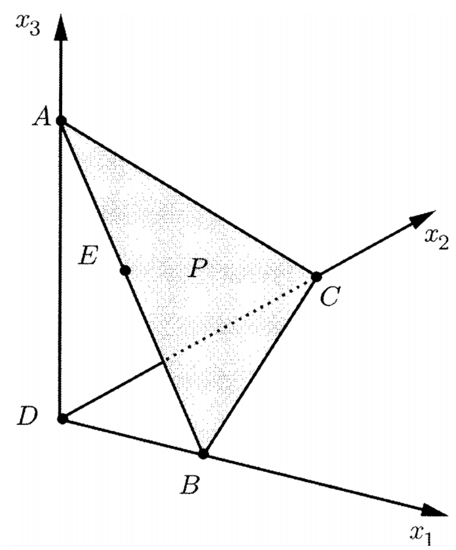
2. *Out of all the constraints that are active at* $\mathbf{x}^*$*, there are* $n$ *of them that are linearly independent.*

*In addition,* $\mathbf{x}^*$ *is said to be a* basic feasible solution *if it is a basic solution and satisfies all the constraints, i.e.,* $\mathbf{x}^* \in P$.

**Remark 2.14 (degeneracy)** *Note that the definition of basic solution only requires there are $n$ linearly independent active constraints. While this implies the number of active constraints has to be greater than or equal to $n$, in case there are more than $n$ constraints active at a basic solution, we call it a* degenerate basic solution.

**Remark 2.15 (adjacency)** *Two distinct basic solutions are said to be* adjacent *if we can find $n-1$ linearly independent constraints that are active at both of them.*

**Example 2.16** *Let $P = \{(x_1, x_2, x_3) | x_1 + x_2 + x_3 = 1, x_1, x_2, x_3 \geq 0\}$. The figure below gives an illustration of the polyhedron $P$. Among the five points $A, B, C, D, E$ which are basic solutions, basic feasible solutions, degenerate basic solutions respectively?*




**Example 2.17** *Let $P = \{(x_1, x_2, x_3) | x_1 + x_2 + x_3 \leq 1, x_1 + x_2 + x_3 \geq 1, x_1, x_2, x_3 \geq 0\}$. The same figure above describes the polyhedron $P$. Among the five points $A, B, C, D, E$ which are basic solutions, basic feasible solutions, degenerate basic solutions respectively?*

**Remark 2.18** *The concept of basic solution and degeneracy are not purely geometric properties. That is, for the same point in the same polyhedron, it can be a basic solution in one algebraic representation of the polyhedra and not a basic solution in another. Basic feasible solution, on the other hand, is a pure geometric property as the following theorem shows, even though its definition relies on a specific representation of the polyhedron.*

**Theorem 2.19** *Let $P$ be a nonempty polyhedron and let $\mathbf{x}^* \in P$. Then the following are equivalent:*

**(a)** $\mathbf{x}^*$ *is a vertex;*

**(b)** $\mathbf{x}^*$ *is an extreme point;*

**(c)** $\mathbf{x}^*$ *is a basic feasible solution.*

*Proof:* Without loss of generality, we assume that

$$P = \left\{ \mathbf{x} \in \mathbb{R}^n \;\middle|\; \begin{array}{lll} \mathbf{a}_i'\mathbf{x} & \geq b_i, & i \in M_1 \\ \mathbf{a}_i'\mathbf{x} & = b_i, & i \in M_2. \end{array} \right\}.$$

(a)$\Rightarrow$(b): Suppose $\mathbf{x}^*$ is a vertex, then by definition there exists $\mathbf{c}$ such that for any $\mathbf{y}, \mathbf{z} \in P$ and $\mathbf{y} \neq \mathbf{x}^*, \mathbf{z} \neq \mathbf{x}^*$, we have $\mathbf{c}'\mathbf{x}^* < \mathbf{c}'\mathbf{y}, \mathbf{c}'\mathbf{x}^* < \mathbf{c}'\mathbf{z}$. Hence, for any $0 \leq \lambda \leq 1$, we have

$$\mathbf{c}'\mathbf{x}^* < \mathbf{c}'(\lambda\mathbf{y} + (\mathbf{1} - \lambda)\mathbf{z}),$$

and consequently $\mathbf{x}^* \neq \lambda\mathbf{y} + (\mathbf{1} - \lambda)\mathbf{z}$. By definition $\mathbf{x}^*$ is an extreme point.

(b)$\Rightarrow$(c): Suppose $\mathbf{x}^*$ is not a basic feasible solution. Let $I = \{i | \mathbf{a}_i'\mathbf{x}^* = b_i\}$ and consider the matrix $\mathbf{A}$ formed with the rows being $\mathbf{a}_i', i \in I$. Since $\mathbf{x}^*$ is not a BFS, one can not find $n$ linearly independent vectors in the family $\mathbf{a}_i, i \in I$. As a result, the matrix $\mathbf{A}$ has rank less than $n$ and one can find $\mathbf{d} \in \mathbb{R}^n, \mathbf{d} \neq \mathbf{0}$ such that $\mathbf{Ad} = \mathbf{0}$ or equivalently $\mathbf{a}_i'\mathbf{d} = 0, i \in I$. Let $\epsilon$ be a small positive number and $\mathbf{y} = \mathbf{x}^* + \epsilon\mathbf{d}$, $\mathbf{z} = \mathbf{x}^* - \epsilon\mathbf{d}$. We then have $\mathbf{a}_i'\mathbf{y} = \mathbf{a}_i'\mathbf{x}^* = b_i$ for $i \in I$. In addition, for $i \notin I$, we have $\mathbf{a}_i'\mathbf{y} = \mathbf{a}_i'\mathbf{x}^* + \epsilon\mathbf{a}_i'\mathbf{d} > b_i$, provided that $\epsilon$ is chosen to be sufficiently small. Hence, $\mathbf{y} \in P$. One can similarly argue that there exists $\epsilon$ sufficiently small such that $\mathbf{z} \in P$. The construction above, however, implies that

$$\mathbf{x}^* = \frac{\mathbf{y} + \mathbf{z}}{2},$$

which implies $\mathbf{x}^*$ is not an extreme point.

(c)$\Rightarrow$(a): Let $\mathbf{x}^*$ be a basic feasible solution and let $I = \{i | \mathbf{a}_i'\mathbf{x}^* = b_i\}$. Consider $\mathbf{c} = \sum_{i \in I} \mathbf{a}_i$. We then have

$$\mathbf{c}'\mathbf{x}^* = \sum_{i \in I} \mathbf{a}_i'\mathbf{x}^* = \sum_{i \in I} b_i.$$

For any $\mathbf{x} \in P$, by feasibility, we know

$$\mathbf{c}'\mathbf{x} = \sum_{i \in I} \mathbf{a}_i'\mathbf{x} \geq \sum_{i \in I} b_i.$$

This shows that $\mathbf{x}^*$ is an optimal solution to the LP

$$\begin{array}{rl} \min & \mathbf{c}'\mathbf{x} \\ \text{s.t.} & \mathbf{x} \in P. \end{array}$$

To show that it is the unique optimal solution, note that a feasible solution $\mathbf{x}$ is optimal if and only if $\mathbf{a}_i'\mathbf{x} = b_i$ for all $i \in I$. From the fact that $\mathbf{x}^*$ is a basic feasible solution, we know that there exist $n$ linearly independent vectors in $\{\mathbf{a}_i', i \in I\}$ which implies that the system of linear equations $\mathbf{a}_i'\mathbf{x} = b_i, i \in I$ must have a unique solution. This proves the claim. ■

## Polyhedra in Standard Form

Now we specialize the above definitions to the case of polyhedra in standard form: $P = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, where $\mathbf{A}$ is an $m \times n$ matrix. Without loss of generality, we assume that $m \leq n$ and the $m$ rows of $\mathbf{A}$ are linearly independent.

To get a basic solution, our goal is to find $n$ linear independent constraints. Since equality constraints have to be satisfied, $\mathbf{Ax} = \mathbf{b}$ gives $m$ such linear independent constraints. It is sufficient to choose $n - m$ of the variables $x_i$ to be 0 and in the mean time keep the constraints

$$\begin{aligned} &\mathbf{Ax} = \mathbf{b}, \\ &x_i = 0, \quad i \in S, \quad |S| = n - m \end{aligned}$$

all linearly independent.

The following result summarizes the above characterization and suggests a way of finding the variables $x_i$ that need to be 0.

**Theorem 2.20** *A vector* $\mathbf{x} \in \mathbb{R}^n$ *is a basic solution if and only if we have* $\mathbf{A}\mathbf{x} = \mathbf{b}$*, and there exist indices* $B(1), ..., B(m)$ *such that:*

**(a)** *The columns* $\mathbf{A}_{B(1)}, ..., \mathbf{A}_{B(m)}$ *are linearly independent;*

**(b)** *If* $i \neq B(1), ..., B(m)$*, then* $x_i = 0$*.*

*Proof:* "if part": Suppose for some $\mathbf{x}$ such that $\mathbf{A}\mathbf{x} = \mathbf{b}$, there exist indices $B(1), ..., B(m)$ such that (a) and (b) hold. It is then sufficient to show that the system of $n$ constraints

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \mathbf{b}, \\ x_i &= 0, \quad i \neq B(1), ..., B(m) \end{aligned} \tag{2.1}$$

are linearly independent. Indeed, given $x_i = 0$ for $i \neq B(1), ..., B(m)$, the equality constraints $\mathbf{A}\mathbf{x} = \sum_{i=1}^n A_i x_i = \mathbf{b}$ reduces to

$$\sum_{i=1}^m A_{B(i)} x_{B(i)} = \mathbf{b},$$

which is a system of $m$ equations with $m$ variables. By property (a), this system of equation has a unique solution which implies that (2.1) also has a unique solution and hence its constraints are linearly independent.

"only if part": Suppose now that $\mathbf{x}$ is a basic solution. Then there exist indices $B(1), ..., B(m)$ such that (2.1) is a system of linearly independent equations. This directly establishes property (b). Again, note that the linear independence of (2.1) implies uniqueness of its solution which in turn implies the uniqueness of the sub-system

$$\sum_{i=1}^m A_{B(i)} x_{B(i)} = \mathbf{b}.$$

Hence, the columns $\mathbf{A}_{B(1)}, ..., \mathbf{A}_{B(m)}$ must be linearly independent. ■

The algebraic characterization of basic solution allows us to find the extreme point (basic feasible solution) according to the following procedure.

1. Choose $m$ linearly independent columns $\mathbf{A}_{B(1)}, ..., \mathbf{A}_{B(m)}$ from the matrix $\mathbf{A}$.
2. Let $x_i = 0$ if $i \neq B(1), ..., B(m)$.
3. Solve the system of $m$ equations $\mathbf{A}\mathbf{x} = \mathbf{b}$ for the unknowns $x_{B(1)}, ..., x_{B(m)}$.
4. If $x_i, 1 \leq i \leq n$ are all non-negative (satisfying the constraints $\mathbf{x} \geq \mathbf{0}$), then we have found a basic feasible solution. Otherwise, repeat step 1 with another set of $m$ linearly independent columns.

We call $x_{B(1)}, ..., x_{B(m)}$ in a basic solution $\mathbf{x}$ the *basic variables* (the remaining are *nonbasic*). The columns $\mathbf{A}_{B(1)}, ..., \mathbf{A}_{B(m)}$ are called the *basic columns* and they form a basis of $\mathbb{R}^m$. We denote the submatrix of $\mathbf{A}$ formed by putting all the basic columns together as

$$\mathbf{B} = \begin{bmatrix} \mathbf{A}_{B(1)} & \mathbf{A}_{B(2)} & ... & \mathbf{A}_{B(m)} \end{bmatrix},$$

called the *basis matrix* and the basic variables

$$\mathbf{x}_B = \begin{bmatrix} x_{B(1)} \\ \vdots \\ x_{B(m)} \end{bmatrix},$$

can be solved as $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$.

What if some of the $x_{B(j)}, 1 \le j \le m$ happens to be 0? Recall the definition of degenerate basic solution. For a standard form problem, a basic solution $\mathbf{x}$ is called degenerate if more than $n - m$ of the components of $\mathbf{x}$ are zero.

**Example 2.21** *Let the constraint* $\mathbf{A}\mathbf{x} = \mathbf{b}$ *be of the form*

$$\begin{bmatrix} 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 6 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 8 \\ 12 \\ 4 \\ 6 \end{bmatrix}.$$

*Clearly,* $\mathbf{A}_4, \mathbf{A}_5, \mathbf{A}_6, \mathbf{A}_7$ *are linearly independent and can serve as basic columns. It is easy to obtain that the resulting basic solution is* $\mathbf{x} = (0, 0, 0, 8, 12, 4, 6)$ *and this is a basic feasible solution.*

*The columns* $\mathbf{A}_3, \mathbf{A}_5, \mathbf{A}_6, \mathbf{A}_7$ *are also linearly independent and lead to basic columns. The resulting basic solution is* $\mathbf{x} = (0, 0, 4, 0, -12, 4, 6)$*, which is not feasible.*

*What if we choose* $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_7$ *as basic columns (they are also linearly independent)? We then have* $x_4 = x_5 = x_6 = 0$ *and the system*

$$\begin{bmatrix} 1 & 1 & 2 & 0 \\ 0 & 1 & 6 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_7 \end{bmatrix} = \begin{bmatrix} 8 \\ 12 \\ 4 \\ 6 \end{bmatrix}.$$

*We can solve* $(x_1, x_2, x_3, x_7) = (4, 0, 2, 6)$ *and* $\mathbf{x} = (4, 0, 2, 0, 0, 0, 6)$*. This is a degenerate basic feasible solution.*

*If we replace the basic column* $\mathbf{A}_2$ *by column* $\mathbf{A}_4$*, it can be verified that* $\mathbf{A}_1, \mathbf{A}_3, \mathbf{A}_4, \mathbf{A}_7$ *are still basic columns and the corresponding basic solution is still* $\mathbf{x} = (4, 0, 2, 0, 0, 0, 6)$.

*Recall the procedure in finding the basic feasible solution. Whenever we change the basic columns, we want to change the basic solution (to a new solution). However, the bad thing about degeneracy is that we might get stuck at the same solution (think about the geometric intuition).*

In standard form, two distinct basic solutions are adjacent if their basis are adjacent—they share all but one basic column. Yet, adjacent basis not necessarily lead to adjacent basic solutions. In the above example, $\{\mathbf{A}_4, \mathbf{A}_5, \mathbf{A}_6, \mathbf{A}_7\}$ and $\{\mathbf{A}_3, \mathbf{A}_5, \mathbf{A}_6, \mathbf{A}_7\}$ are adjacent basis and the corresponding basic solutions $\mathbf{x} = (0, 0, 0, 8, 12, 4, 6)$ and $\mathbf{x} = (0, 0, 4, 0, -12, 4, 6)$ are adjacent. On the other hand, $\{\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_7\}$ and $\{\mathbf{A}_1, \mathbf{A}_3, \mathbf{A}_4, \mathbf{A}_7\}$ are also adjacent, they lead to the same basic solution.

**Example 2.22** *Consider the standard form polyhedron:*

$$P = \{(x_1, x_2, x_3) | x_1 - x_2 = 0, x_1 + x_2 + 2x_3 = 2, x_1, x_2, x_3 \ge 0\}$$

*What are the basic feasible solutions? Which one is degenerate?*

*Now consider the following polyhedron, which is not in standard form.*

$$P = \{(x_1, x_2, x_3)|x_1 - x_2 = 0, x_1 + x_2 + 2x_3 = 2, x_1, x_3 \geq 0\}$$

*What are the basic feasible solutions? Which one is degenerate?*

## 2.5 Existence and Optimality of Extreme Point

Given a polyhedron $P$, does an extreme point always exist? Consider, for example, the polyhedron

$$P = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{A}\mathbf{x} \geq \mathbf{b}\}.$$

What if the number of rows of $\mathbf{A}$ is less than $n$?

**Theorem 2.23** *Suppose that the polyhedron $P = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{A}\mathbf{x} \geq \mathbf{b}\}$ is nonempty. Then $P$ has at least one extreme point if and only if $P$ does not contain a line, that is, there does not exist $\mathbf{x} \in P$ and $\mathbf{d} \in \mathbb{R}^n, \mathbf{d} \neq \mathbf{0}$ such that $\mathbf{x} + \lambda\mathbf{d} \in P$ for any $\lambda \in \mathbb{R}$.*

*Proof*: "if part": For any $\mathbf{x} \in P$, let $I = \{i|\mathbf{a}_i'\mathbf{x} = b_i\}$. If there exist $n$ linearly independent vectors in $\mathbf{a}_i, i \in I$, then by definition $\mathbf{x}$ is an extreme point and we are done. Otherwise the rank for the vectors $\mathbf{a}_i, i \in I$ must be less than $n$, and there exists $\mathbf{d}$ such that $\mathbf{a}_i'\mathbf{d} = 0$ for any $i \in I$. We can construct a line by

$$\mathbf{y} = \mathbf{x} + \lambda\mathbf{d}.$$

Note that for any point on the line and any $i \in I$, we have $\mathbf{a}_i'(\mathbf{x} + \lambda\mathbf{d}) = \mathbf{a}_i'\mathbf{x} = b_i$. On the other hand, since $P$ does not contain a line, when we change $\lambda$, there must exist some $j \notin I$, such that the constraint $\mathbf{a}_j'(\mathbf{x} + \lambda\mathbf{d}) \geq b_j$ is violated. In other words, there exists $j \notin I$ and $\lambda^*$ such that $\mathbf{x} + \lambda^*\mathbf{d} \in P$ and $\mathbf{a}_j'(\mathbf{x} + \lambda^*\mathbf{d}) = b_j$.

Furthermore, $\mathbf{a}_j$ is not a linear combination of $\mathbf{a}_i, i \in I$ since if $\mathbf{a}_j = \sum_{i\in I} \alpha_i\mathbf{a}_i$, we then have $\mathbf{a}_j'\mathbf{d} = \sum_{i\in I} \alpha_i\mathbf{a}_i'\mathbf{d} = 0$. However, from $\mathbf{a}_j'(\mathbf{x} + \lambda^*\mathbf{d}) = b_j$, we know $\mathbf{a}_j'\mathbf{d}$ cannot be zero since $\mathbf{a}_j'\mathbf{x} > b_j$ by $j \notin I$. As a result, we have reached a new point $\mathbf{x} + \lambda^*\mathbf{d} \in P$, for which the rank of the binding constraints increased by one.

The above argument can be repeated until we have reached a point whose binding constraints has rank $n$, i.e., $n$ linearly independent binding constraints—implying the existence of an extreme point.

"only if part": Suppose $\mathbf{x}$ is an extreme point, then there exists $n$ linearly independent binding constraints, say, $\mathbf{a}_1, ..., \mathbf{a}_n$. If $P$ contains a line, say, $\mathbf{x} + \lambda\mathbf{d} \in P$ for some $\mathbf{d} \neq \mathbf{0}$ and for any $\lambda \in \mathbf{R}$. By definition of $P$, we have

$$\mathbf{a}_i'(\mathbf{x} + \lambda\mathbf{d}) \geq b_i, i = 1, ..., n.$$

The above inequality implies $\mathbf{a}_i'\mathbf{d} = 0$ since if $\mathbf{a}_i'\mathbf{d} > 0$, then as $\lambda \to -\infty$, the inequality must be violated (similar argument applies to $\mathbf{a}_i'\mathbf{d} < 0$). Yet, linear independence of $\mathbf{a}_1, ..., \mathbf{a}_n$ implies $\mathbf{d} = \mathbf{0}$, leading to a contradiction. ■

Note that a polyhedron in standard form does not contain a line and therefore has at least one extreme point.

Now we turn to the central question: can we always find an optimal solution within the set of extreme points of the feasible set? The following theorem provides an affirmative answer.

**Theorem 2.24** *Consider the linear programming problem of minimizing* $\mathbf{c}'\mathbf{x}$ *over a polyhedron* $P$. *Suppose that* $P$ *has at least one extreme point and there exists an optimal solution. Then, there exists an optimal solution which is an extreme point of* $P$.

*Proof:* Let $P$ be of the form $P = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{A}\mathbf{x} \geq \mathbf{b}\}$, $Q$ be the set of optimal solutions and $v$ be the optimal value. Then $Q$ can be represented as

$$Q = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{A}\mathbf{x} \geq \mathbf{b}, \mathbf{c}'\mathbf{x} = v\}.$$

Note that $Q \subset P$ is a polyhedron and has at least one extreme point (why?). Let $\mathbf{x}^*$ be an extreme point of $Q$. What remains is to show that $\mathbf{x}^*$ is also an extreme point of $P$.

We prove by contradiction. Suppose $\mathbf{x}^*$ is not an extreme point of $P$, then there exists $\mathbf{y}, \mathbf{z} \in P, \mathbf{y}, \mathbf{z} \neq \mathbf{x}$ and $\lambda \in [0, 1]$ such that $\mathbf{x} = \lambda\mathbf{y} + (1 - \lambda)\mathbf{z}$. It follows that $v = \mathbf{c}'\mathbf{x} = \lambda\mathbf{c}'\mathbf{y} + (1 - \lambda)\mathbf{c}'\mathbf{z}$. Since $v$ is the minimum cost, we must have $\mathbf{c}'\mathbf{y} \geq v$ and $\mathbf{c}'\mathbf{z} \geq v$. Thus, the only way for $\lambda\mathbf{c}'\mathbf{y} + (1 - \lambda)\mathbf{c}'\mathbf{z} = v$ is $\mathbf{c}'\mathbf{y} = \mathbf{c}'\mathbf{z} = v$, which implies $\mathbf{y}, \mathbf{z} \in Q$. However, this contradicts with the fact that $\mathbf{x}^*$ is an extreme point in $Q$. ■

In fact, we have a stronger claim that an extreme point is an optimal solution as long as the optimal cost is finite.

**Theorem 2.25** *Consider the linear programming problem of minimizing* $\mathbf{c}'\mathbf{x}$ *over a polyhedron* $P$. *Suppose that* $P$ *has at least one extreme point. Then, either the optimal cost is equal to* $-\infty$, *or there exists an extreme point which is optimal.*

*Proof:* We assume that the optimal cost is finite and let $P$ be of the form $P = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{A}\mathbf{x} \geq \mathbf{b}\}$. Given any $\mathbf{x} \in P$, let $I = \{i | \mathbf{a}_i'\mathbf{x} = b_i\}$. Suppose that the rank of $\mathbf{a}_i, i \in I$ is less than $n$, i.e., $\mathbf{x}$ is not an extreme point. We show in the following that one can always an extreme point such that it yields a cost at most $\mathbf{c}'\mathbf{x}$.

The rank of $\mathbf{a}_i, i \in I$ being less than $n$ implies we can find $\mathbf{d}$ such that $\mathbf{a}_i'\mathbf{d} = 0$ for every $i \in I$. In addition, by appropriately choosing the sign for $d_i$, we can always make $\mathbf{c}'\mathbf{d} \leq 0$ without affecting the constraints $\mathbf{a}_i'\mathbf{d} = 0$ for $i \in I$.

Suppose we can find $\mathbf{d}$ such that $\mathbf{c}'\mathbf{d} < 0$. Let $\mathbf{y} = \mathbf{x} + \lambda\mathbf{d}$, where $\lambda$ is positive. Note that $\mathbf{c}'(\mathbf{x} + \lambda\mathbf{d}) < \mathbf{c}'\mathbf{x}$. If the half-line is contained in $P$, then the optimal cost can be decreased all the way to $-\infty$. Otherwise, as shown before, one can find $\lambda^*$ and $j \notin I$ such that $\mathbf{a}_i'(\mathbf{x} + \lambda^*\mathbf{d}) = b_j$ with $\mathbf{x} + \lambda^*\mathbf{d} \in P$ and the rank of its binding constraints increased by one.

Suppose that $\mathbf{c}'\mathbf{d} = 0$. We can then consider the line $\mathbf{y} = \mathbf{x} + \lambda\mathbf{d}$, with $\lambda \in \mathbb{R}$. Since an extreme point exists, $P$ does not contain a line, and we can find some $\mathbf{y}$ such that the rank of its binding constrains is increased by one.

In either case, we have found a new point $\mathbf{y}$, whose binding constraints have higher rank than $\mathbf{x}$ and $\mathbf{c}'\mathbf{y} \leq \mathbf{c}'\mathbf{x}$. Repeating the process, we can then find an extreme point $\mathbf{w}$ such that $\mathbf{c}'\mathbf{w} \leq \mathbf{c}'\mathbf{x}$. ■

Note that for general optimization problems, a finite optimal cost does not necessarily imply the existence of optimal solution as shown by the following example.

$$\begin{aligned} \min \quad & 1/x \\ \text{s.t.} \quad & x \geq 1. \end{aligned}$$

# References

[BT97] D. Bertsimas and J.N. Tsitsiklis, *Introduction to Linear Optimization*, Springer, 1997.

**BDC6111: Introduction to Optimization** **Fall 2018, NUS**

# Lecture 3: Simplex Method

*Lecturer: Zhenyu Hu*

A general idea in many optimization algorithms is as follows:

- Find a feasible solution.
- Search its neighborhood to find a nearby feasible solution with lower cost.
- If no nearby feasible solution leads to lower cost, then the algorithm terminates and we have a *local optimal* solution.

For linear programming problems, we have seen in the lecture "Geometry of Linear Program" that a basic feasible solution is an ideal feasible solution to start with and when considering a neighborhood, it is sufficient to consider the nearby basic feasible solutions. In addition, for linear programming problems, local optimal solution is always a global optimal solution. The only remaining question is how to go from one basic feasible solution to another one.

In the following, we consider the standard form problem

$$\begin{aligned} \min \quad & \mathbf{c}'\mathbf{x} \\ s.t. \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

We let $P$ denote the corresponding feasible set and we assume that the rows of $\mathbf{A}$ are linear independent.

## 3.1 Searching Direction and Optimality Condition

Suppose we are sitting at $\mathbf{x} \in P$ and we are considering moving away from $\mathbf{x}$ along the direction of $\mathbf{d} \in \mathbb{R}^n$. We should move along the direction that satisfies the following considerations:

- It does not immediately lead us outside of $P$, i.e., it leads to a feasible solution.
- It is an edge of $P$.
- Along the direction, the cost is reduced.

For feasibility, we need the following definition.

**Definition 3.1** *A vector $\mathbf{d} \in \mathbb{R}^n$ is said to be a* feasible direction *at $\mathbf{x}$, if there exists a positive scalar $\theta$ for which $\mathbf{x} + \theta\mathbf{d} \in P$.*

As we mentioned, $\mathbf{x}$ is usually taken as a basic feasible solution with $B(1), ..., B(m)$ be the indices of the basic variables and

$$\mathbf{B} = \begin{bmatrix} \mathbf{A}_{B(1)} & \mathbf{A}_{B(2)} & \dots & \mathbf{A}_{B(m)} \end{bmatrix},$$

being the basis matrix. We know that the basic variables in $\mathbf{x}$ are $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$ and the non-basic variables are all 0. In addition, for adjacent BFS, they share all but one basic column.

To move to an adjacent BFS, the direction $\mathbf{d}$ must be of the form $d_j = 1$ and $d_i = 0$ for some *nonbasic index* $j$ and every *nonbasic index* $i$ other than $j$. Feasibility further indicates $\mathbf{A}(\mathbf{x}+\theta\mathbf{d}) = \mathbf{b}$ and $\mathbf{x}+\theta\mathbf{d} \geq 0$ for some $\theta > 0$.

For equality constraints, we have $\mathbf{A}(\mathbf{x}+\theta\mathbf{d}) = \mathbf{b} = \mathbf{A}\mathbf{x}+\theta\mathbf{A}\mathbf{d} = \mathbf{b}+\theta\mathbf{A}\mathbf{d} = \mathbf{b}$. That is, we require $\mathbf{A}\mathbf{d} = \mathbf{0}$. It follows that

$$\mathbf{A}\mathbf{d} = \sum_{i=1}^{n} \mathbf{A}_i d_i = \sum_{i=1}^{m} \mathbf{A}_{B(i)} d_{B(i)} + \mathbf{A}_j = \mathbf{B}\mathbf{d}_B + \mathbf{A}_j = \mathbf{0},$$

where $\mathbf{d}_B = (d_{B(1)}, ..., d_{B(m)})$ and we can obtain

$$\mathbf{d}_B = -\mathbf{B}^{-1}\mathbf{A}_j.$$

For non-negativity constraints, if $\mathbf{x}$ is a non-degenerate basic feasible solution, we then have $\mathbf{x}_B > \mathbf{0}$. This implies $\mathbf{x}_B + \theta\mathbf{d}_B \geq \mathbf{0}$ for sufficiently small $\theta$. For nonbasic variables, we know that $x_j + \theta d_j = \theta \geq 0$ and $x_i + \theta d_i = x_i = 0$ for all $i \neq j$. That is, $\mathbf{x}+\theta\mathbf{d} \geq 0$ hold for $\theta$ small. What if $\mathbf{x}$ is degenerate?

Finally, we want a direction that reduces the cost. In other words, we want to compare $\mathbf{c}'\mathbf{x}$ and $\mathbf{c}'\mathbf{x}+\theta\mathbf{c}'\mathbf{d}$. The rate of cost change (difference divide by $\theta$) along this direction is

$$\mathbf{c}'\mathbf{d} = \mathbf{c}'_B\mathbf{d}_B + c_j = c_j - \mathbf{c}'_B\mathbf{B}^{-1}\mathbf{A}_j,$$

where $\mathbf{c}'_B = (c_{B(1)}, ..., c_{B(m)})$. The above quantity is called the *reduced cost* of the variable $x_j$ and we denote it as $\bar{c}_j = c_j - \mathbf{c}'_B\mathbf{B}^{-1}\mathbf{A}_j$.

**Example 3.2** *Consider the linear programming problem*

$$\begin{array}{ll} \min & 2x_1 + 2x_2 + x_3 + x_4 \\ s.t. & \begin{array}{rrrrl} x_1+ & x_2+ & x_3+ & x_4 & = 2 \\ 2x_1+ & & 3x_3+ & 4x_4 & = 2 \end{array} \\ & x_1, x_2, x_3, x_4 \geq 0. \end{array}$$

*Clearly,* $\mathbf{A}_1 = (1,2), \mathbf{A}_2 = (1,0)$ *are linearly independent and can serve as basic columns. It is easy to obtain that the resulting basic solution is* $\mathbf{x} = (1,1,0,0)$ *and this is a basic feasible solution.*

*To obtain a searching direction, we let* $d_3 = 1$ *and* $d_4 = 0$*. Then the direction of change of the basic variables is*

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \mathbf{d}_B = -\mathbf{B}^{-1}\mathbf{A}_3 = -\begin{bmatrix} 0 & 1/2 \\ 1 & -1/2 \end{bmatrix}\begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} -3/2 \\ 1/2 \end{bmatrix}$$

*The reduced cost is then*

$$\bar{c}_3 = c_3 - [c_1, c_2]\mathbf{B}^{-1}\mathbf{A}_3 = 1 + [2,2]\begin{bmatrix} -3/2 \\ 1/2 \end{bmatrix} = -1.$$

*We can also directly verify this. Note that* $\mathbf{c}'\mathbf{x} = 4$*. The new solution we find by moving one unit along the direction* $\mathbf{d}$ *is* $\mathbf{x}+\mathbf{d} = (-1/2, 3/2, 1, 0)$*. One can check that* $\mathbf{c}'(\mathbf{x}+\mathbf{d}) = 3$*. The cost is reduced by* 1*. Unfortunately, the new solution we find* $(-1/2, 3/2, 1, 0)$ *is not feasible because we have went too far along the direction* $\mathbf{d}$*.*

The optimality condition for LP is then summarized in the following theorem.

**Theorem 3.3** *Consider a BFS* $\mathbf{x}$ *associated with a basis matrix* $\mathbf{B}$*, and let* $\bar{\mathbf{c}}$ *be the corresponding vector of reduced costs.*

*(a) If* $\bar{\mathbf{c}} \geq \mathbf{0}$*, then* $\mathbf{x}$ *is optimal.*

*(b) If* $\mathbf{x}$ *is optimal and nondegenerate, then* $\bar{\mathbf{c}} \geq \mathbf{0}$*.*

*Proof:* (a) Let $\mathbf{y}$ be any feasible solution, and let $\mathbf{d} = \mathbf{y} - \mathbf{x}$. By feasibility of $\mathbf{x}, \mathbf{y}$, we have $\mathbf{Ad} = \mathbf{A}(\mathbf{y} - \mathbf{x}) = \mathbf{0}$, which is equivalent as

$$\mathbf{B}\mathbf{d}_{\mathbf{B}} + \sum_{i \in N} \mathbf{A}_i d_i = \mathbf{0},$$

where $N$ is the set of nonbasic indices. It follows that $\mathbf{d}_{\mathbf{B}} = -\sum_{i \in N} \mathbf{B}^{-1}\mathbf{A}_i d_i$, and

$$\mathbf{c}'\mathbf{d} = \mathbf{c}'_{\mathbf{B}}\mathbf{d}_{\mathbf{B}} + \sum_{i \in N} c_i d_i = \sum_{i \in N} (c_i - \mathbf{c}'_{\mathbf{B}}\mathbf{B}^{-1}\mathbf{A}_i) d_i = \sum_{i \in N} \bar{c}_i d_i.$$

Note that for $i \in N$, we have $x_i = 0$, and hence $d_i = y_i \geq 0$, and by $\bar{c}_i \geq 0$, we have $\mathbf{c}'\mathbf{d} \geq 0$.

(b) Suppose on the contrary, $\bar{c}_j < 0$. Then $j$ must correspond to a nonbasic variable. The nondegeneracy of $\mathbf{x}$ implies we can find a $\theta > 0$ and $\mathbf{d}$ with $d_j = 1$, $d_i = 0$ for any nonbasic index $i, i \neq j$ such that $\mathbf{x} + \theta\mathbf{d}$ is feasible. In addition, the reduced cost implies the cost reduces by $\theta\bar{c}_j$ by moving along the direction $\mathbf{d}$. ■

Following the above theorem, any basis matrix $\mathbf{B}$ is called optimal if (a) $\mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$ and (b) $\bar{\mathbf{c}} \geq \mathbf{0}$.

## 3.2 Simplex Method

Following the Example 3.2, suppose we have computed the reduced cost $\bar{c}_j$ is negative for the nonbasic variable $x_j$ (in the above example, it is $x_3$). The reduced cost tells us it is better to make $x_j$ positive and we call this situation that we want to *bring* $x_j$ *into the basis* or $x_j$ *should enter the basis*. The question now is how positive should $x_j$ be or in other words, how far shall we go along the direction $\mathbf{d}$. We have seen in the above example that if we go too far, $\mathbf{x} + \theta\mathbf{d}$ will become infeasible. On the other hand, the cost reduced by $\mathbf{x} + \theta\mathbf{d}$ is $\theta\bar{c}_j$, which means the larger $\theta$ is the better. This takes us to the point $\mathbf{x} + \theta^*\mathbf{d}$, where

$$\theta^* = \max\{\theta \geq 0 | \mathbf{x} + \theta\mathbf{d} \in P\}.$$

Recall that $\mathbf{d}$ is chosen such that $\mathbf{Ad} = 0$. Thus, the equality constraints $\mathbf{A}(\mathbf{x} + \theta\mathbf{d}) = \mathbf{b}$ will always be satisfied for any $\theta \geq 0$. We only need to worry about the non-negativity constraints. There are two cases:

1. If $\mathbf{d} \geq \mathbf{0}$, then $\mathbf{x} + \theta\mathbf{d} \geq \mathbf{0}$ for any $\theta \geq 0$. It is optimal to choose $\theta^* = \infty$.
2. If $d_i < 0$ for some $i$, the constraint $x_i + \theta d_i \geq 0$ becomes $\theta \leq -x_i/d_i$. Thus,

$$\theta^* = \min_{\{i | d_i < 0\}} \left( -\frac{x_i}{d_i} \right).$$

**Remark 3.4** *Note that if* $x_i$ *is a nonbasic variable, then either* $d_i = 1$ *or* $d_i = 0$*. Thus, it is sufficient to consider the basic variables when finding* $\theta^*$*, i.e.,*

$$\theta^* = \min_{\{i = 1, \ldots, m | d_{B(i)} < 0\}} \left( -\frac{x_{B(i)}}{d_{B(i)}} \right).$$

*If $\mathbf{x}$ is a non-degenerate basic feasible solution, then $x_{B(i)} > 0$ for all $i = 1, ..., m$ which implies $\theta^* > 0$. This means that we are guaranteed to move to a new point. On the other hand, if for some $i = 1, ..., m$, it happens that $d_{B(i)} < 0$ and $x_{B(i)} = 0$—in which case $\mathbf{x}$ is degenerate, then $\theta^* = 0$ and we are stuck at the same point.*

**Example 3.5** *We continue with Example 3.2. We can compute $\mathbf{x} + \theta\mathbf{d}$ as $(1 - \frac{3}{2}\theta, 1 + \frac{1}{2}\theta, \theta, 0)$. Clearly, for $\mathbf{x} + \theta\mathbf{d} \geq \mathbf{0}$, we only need $1 - \frac{3}{2}\theta \geq 0$. That is, $\theta^* = \frac{2}{3}$ and $\mathbf{x} + \theta^*\mathbf{d} = (0, \frac{4}{3}, \frac{2}{3}, 0)$.*

*Note that the point $(0, \frac{4}{3}, \frac{2}{3}, 0)$ is another basic feasible solution with the corresponding basic columns being $\mathbf{A}_2 = (1, 0)$ and $\mathbf{A}_3 = (1, 3)$. Thus, we have successfully moved from one basic feasible solution to another one while reducing the cost.*

We now formalize the observation made in Example 3.5. We assume $\mathbf{x}$ is non-degenerate. Suppose $l$ is the index that solves

$$\min_{\{i=1,...,m | d_{B(i)}<0\}} \left( -\frac{x_{B(i)}}{d_{B(i)}} \right).$$

In other words,

$$-\frac{x_{B(l)}}{d_{B(l)}} = \min_{\{i=1,...,m | d_{B(i)}<0\}} \left( -\frac{x_{B(i)}}{d_{B(i)}} \right) = \theta^*.$$

Rearranging the terms above, we have $x_{B(l)} + \theta^* d_{B(l)} = 0$. In summary, in moving from $\mathbf{x}$ to $\mathbf{x} + \theta^*\mathbf{d}$, the nonbasic variable $x_j = 0$ becomes $x_j + \theta^* > 0$ while the basic variable $x_{B(l)} > 0$ becomes $x_{B(l)} + \theta^* d_{B(l)} = 0$. We call this situation as $x_j$ replaces $x_{B(l)}$ in the basis. Accordingly, the old basis matrix $\mathbf{B}$ is replaced by the new matrix $\overline{\mathbf{B}}$ by changing the basic columns $\mathbf{A}_{B(l)}$ by $\mathbf{A}_j$, i.e.,

$$\overline{\mathbf{B}} = \begin{bmatrix} \mathbf{A}_{B(1)} & \dots & \mathbf{A}_{B(l-1)} & \mathbf{A}_j & \mathbf{A}_{B(l+1)} & \dots & \mathbf{A}_{B(m)} \end{bmatrix}.$$

The following result confirms our observation in Example 3.5.

**Theorem 3.6**

*(a) $\overline{\mathbf{B}}$ is a basis matrix. That is, the $m$ columns $\mathbf{A}_{B(1)}, ..., \mathbf{A}_{B(l-1)}, \mathbf{A}_j, \mathbf{A}_{B(l+1)}, ..., \mathbf{A}_{B(m)}$ are linearly independent.*

*(b) $\mathbf{x} + \theta^*\mathbf{d}$ is a basic feasible solution.*

*Proof:* (a) Suppose, on the contrary that $\mathbf{A}_{\overline{B}(i)}, i = 1, ..., m$ are linearly dependent. Then, there exist $\lambda_i, i = 1, ..., m$ not all zero, such that $\sum_{i=1}^m \lambda_i \mathbf{A}_{\overline{B}(i)} = \mathbf{0}$, which implies

$$\sum_{i=1}^m \lambda_i \mathbf{B}^{-1}\mathbf{A}_{\overline{B}(i)} = \mathbf{0},$$

i.e., the vectors $\mathbf{B}^{-1}\mathbf{A}_{\overline{B}(i)}, i = 1, ..., m$ are also linearly dependent.

On the other hand, for $i \neq l$, we have

$$\mathbf{B}^{-1}\mathbf{A}_{\overline{B}(i)} = \mathbf{B}^{-1}\mathbf{A}_{B(i)} = \mathbf{e}_i$$

and for $i = l$, we have

$$\mathbf{B}^{-1}\mathbf{A}_{\overline{B}(l)} = \mathbf{B}^{-1}\mathbf{A}_j = -\mathbf{d}_{\mathbf{B}}.$$

By definition of the index $l$, $d_{B(l)} < 0$ is nonzero and hence $-\mathbf{d}_{\mathbf{B}}$ cannot be represented as a linear combination of $\mathbf{e}_i, i = 1, ..., m, i \neq l$. Therefore, $\mathbf{B}^{-1}\mathbf{A}_{\overline{B}(i)}, i = 1, ..., m$ are linearly independent, a contradiction.

(b) By construction, the vector $\mathbf{y} = \mathbf{x} + \theta^* \mathbf{d}$ satisfies $y_i = 0$ for $i \neq \bar{B}(1), ...\bar{B}(m)$. The linearly independence of $\mathbf{A}_{\overline{B}(i)}$ then uniquely determines $\mathbf{y}$ as the basic solution corresponding to the basis matrix $\bar{\mathbf{B}}$. It is clearly feasible by our construction. ■

We now summarize one iteration in the simplex method, called a *pivot*.

**An iteration of the simplex method**

1. We start with a basic feasible solution $\mathbf{x}$ (assuming non-degenerate) with the basic matrix
$$\mathbf{B} = \begin{bmatrix} \mathbf{A}_{B(1)} & \mathbf{A}_{B(2)} & \dots & \mathbf{A}_{B(m)} \end{bmatrix},$$
2. Compute the reduced costs $\bar{c}_j = c_j - \mathbf{c}'_B \mathbf{B}^{-1} \mathbf{A}_j$ for all nonbasic indices $j$. If they are all nonnegative, the current basic feasible solution is optimal, and the algorithm terminates; else, choose some $j$ for which $\bar{c}_j < 0$.
3. Compute the search direction $\mathbf{d}$ associated with $x_j$: $d_j = 1$, $d_i = 0$ for all nonbasic indices $i \neq j$ and
$$\mathbf{d}_B = -\mathbf{B}^{-1} \mathbf{A}_j.$$
If $\mathbf{d} \geq \mathbf{0}$ ($\mathbf{d}_B \geq \mathbf{0}$), then $\theta^* = \infty$, optimal cost is $-\infty$, and the algorithm terminates.
4. If some component of $\mathbf{d}$ is negative, let
$$\theta^* = \min_{\{i | d_i < 0\}} \left( -\frac{x_i}{d_i} \right).$$
5. Let $l$ be such that $\theta^* = -\frac{x_{B(l)}}{d_{B(l)}}$. We arrive at the new basic feasible solution $\mathbf{x} + \theta^* \mathbf{d}$ and the corresponding new basis: $\overline{\mathbf{B}}$. We are back to the situation in Step 1.

**Remark 3.7** *At each iteration or pivot, if the algorithm does not terminate (either arriving at optimal solution or optimal cost is $-\infty$), we are guaranteed to move to a new basic feasible solution while reducing the cost. Since there are finite number of basic feasible solutions, the algorithm must terminate in finite number of iterations.*

**Issue with degeneracy**

The degeneracy will only cause a problem if for some $d_{B(l)} < 0$, we have $x_{B(l)} = 0$. In this case, we may nevertheless proceed to replace $x_{B(l)}$ by $x_j$ in the basis. This is worthwhile since now we have a new direction $d_{B(l)} = 1$ to explore.

With degeneracy, it is also possible that after multiple pivoting without any cost reduction, one is back to the initial basis—which is called *cycling*. There are rules to avoid such cycling phenomenon, and simplex method is guaranteed to terminate in finite number of iterations under such rules (see Section 3.4 in [BT97]).

## 3.3 Full Tableau Implementation

We have seen in the above summary that in Step 2 and Step 3, we need $\mathbf{B}^{-1}$ in order to compute the corresponding reduced cost and search direction respectively. The information of $\mathbf{B}$ is not directly used in the computation. This suggests that instead of performing the naive computations

$$\mathbf{B} \to \mathbf{B}^{-1} \to \overline{\mathbf{B}} \to \overline{\mathbf{B}}^{-1},$$

we can update only the information that can be directly used:

$$\mathbf{B}^{-1} \to \overline{\mathbf{B}}^{-1}.$$

The full tableau implementation of the simplex method is based on this idea and updates the following big table at each iteration: We explain each term in detail below:

| | |
|---|---|
| $-\mathbf{c}_B'\mathbf{B}^{-1}\mathbf{b}$ | $\mathbf{c}' - \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A}$ |
| $\mathbf{B}^{-1}\mathbf{b}$ | $\mathbf{B}^{-1}\mathbf{A}$ |

- $-\mathbf{c}_B'\mathbf{B}^{-1}\mathbf{b}$: Recall that the basic variable is given by $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$. Thus, the cost at $\mathbf{x}$ is given by

  $$\mathbf{c}'\mathbf{x} = \mathbf{c}_B'\mathbf{x}_B = \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{b},$$

  which implies $-\mathbf{c}_B'\mathbf{B}^{-1}\mathbf{b}$ is the negative of the current cost.
- $\mathbf{c}' - \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A}$: This is a row vector of length $n$. It's $j$-th component is given by $c_j - \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A}_j$, which is the reduced cost along direction $j$.
- $\mathbf{B}^{-1}\mathbf{b}$: As we mentioned above, this is simply the value of basic variables: $\mathbf{x}_B$.
- $\mathbf{B}^{-1}\mathbf{A}$: We use the following special case of $\mathbf{A}$ to illustrate this term. Recall that $\mathbf{A}$ is an $m \times n$ matrix and if we write column-wise:

  $$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \dots & \mathbf{A}_n \end{bmatrix}.$$

  Among the $n$ columns, $m$ are basic columns associated with $\mathbf{x}$ and they form a basis matrix $\mathbf{B}$. We assume that the $m$ basic columns are the last $m$ columns of $\mathbf{A}$. Then $\mathbf{A}$ can be written as

  $$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \dots & \mathbf{A}_{n-m} & \mathbf{B} \end{bmatrix}.$$

  $\mathbf{B}^{-1}\mathbf{A}$ is then simply

  $$\mathbf{B}^{-1}\mathbf{A} = \begin{bmatrix} \mathbf{B}^{-1}\mathbf{A}_1 & \mathbf{B}^{-1}\mathbf{A}_2 & \dots & \mathbf{B}^{-1}\mathbf{A}_{n-m} & \mathbf{I} \end{bmatrix}.$$

  Now, for each nonbasic indices $1 \le j \le n-m$, $\mathbf{B}^{-1}\mathbf{A}_j$ is the negative of $\mathbf{d}_B$ associated with the nonbasic variable $x_j$.

In summary, the tableau can be alternatively written as:

| | | | |
|---|---|---|---|
| $-\mathbf{c}'\mathbf{x}$ | $\bar{c}_1$ | ... | $\bar{c}_n$ |
| $\mathbf{x}_B$ | $\mathbf{B}^{-1}\mathbf{A}_1$ | ... | $\mathbf{B}^{-1}\mathbf{A}_n$ |

To update the tableau, we summarize the steps as follows:

**An iteration of the full tableau implementation**

1. We start with a tableau

| | | | |
|---|---|---|---|
| $-\mathbf{c}'\mathbf{x}$ | $\bar{c}_1$ | ... | $\bar{c}_n$ |
| $\mathbf{x}_B$ | $\mathbf{B}^{-1}\mathbf{A}_1$ | ... | $\mathbf{B}^{-1}\mathbf{A}_n$ |

2. If $\bar{c}_i$ are all nonnegative, the current basic feasible solution is optimal, and the algorithm terminates; else, choose some $j$ for which $\bar{c}_j < 0$.

3. Check the $j$-th column $\mathbf{B}^{-1}\mathbf{A}_j$. We call this column the *pivot column*. If $\mathbf{B}^{-1}\mathbf{A}_j \leq \mathbf{0}$ (i.e., $\mathbf{d}_B \geq \mathbf{0}$), then the optimal cost is $-\infty$, and the algorithm terminates.

4. If some component of $\mathbf{B}^{-1}\mathbf{A}_j$ is positive, let $u_i$ be the $i$-th component of $\mathbf{B}^{-1}\mathbf{A}_j$ (which is simply $-d_i$). Compute $x_{B(i)}/u_i$. Let $l$ be the index such that
$$x_{B(l)}/u_l = \min_{\{i|u_i>0\}} \left(\frac{x_i}{u_i}\right) = \min_{\{i|d_i<0\}} \left(-\frac{x_i}{d_i}\right).$$
The $l$-th row is called the *pivot row*.

5. Add to each row of the tableau a constant multiple of the $l$-th row so that the $j$-th column
$$\begin{bmatrix} \bar{c}_j \\ \mathbf{B}^{-1}\mathbf{A}_j \end{bmatrix}$$
becomes
$$\begin{bmatrix} 0 \\ \mathbf{e}_l \end{bmatrix}.$$
The row operations are equivalent as constructing a matrix such that
$$\begin{bmatrix} 1 & k\mathbf{e}_l' \\ \mathbf{0} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \bar{c}_j \\ \mathbf{B}^{-1}\mathbf{A}_j \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{e}_l \end{bmatrix},$$
where the vector $[1\ k\mathbf{e}_l']$ represents row operations that makes the zeroth row $\bar{c}_j$ to 0 and $\mathbf{Q}$ represents row operations that make $\mathbf{B}^{-1}\mathbf{A}_j$ to $\mathbf{e}_l$.

For such an iteration to achieve what we want, the key step is Step 5: we need to show that after a sequence of row operations, we can indeed update the tableau as
$$\begin{bmatrix} -\mathbf{c}_B'\mathbf{B}^{-1}\mathbf{b} & \mathbf{c}' - \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A} \\ \mathbf{B}^{-1}\mathbf{b} & \mathbf{B}^{-1}\mathbf{A} \end{bmatrix} \to \begin{bmatrix} -\mathbf{c}_{\overline{B}}'\overline{\mathbf{B}}^{-1}\mathbf{b} & \mathbf{c}' - \mathbf{c}_{\overline{B}}'\overline{\mathbf{B}}^{-1}\mathbf{A} \\ \overline{\mathbf{B}}^{-1}\mathbf{b} & \overline{\mathbf{B}}^{-1}\mathbf{A} \end{bmatrix},$$
or equivalently
$$\begin{bmatrix} -\mathbf{c}_{\overline{B}}'\overline{\mathbf{B}}^{-1}\mathbf{b} & \mathbf{c}' - \mathbf{c}_{\overline{B}}'\overline{\mathbf{B}}^{-1}\mathbf{A} \\ \overline{\mathbf{B}}^{-1}\mathbf{b} & \overline{\mathbf{B}}^{-1}\mathbf{A} \end{bmatrix} = \begin{bmatrix} 1 & k\mathbf{e}_l' \\ \mathbf{0} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} -\mathbf{c}_B'\mathbf{B}^{-1}\mathbf{b} & \mathbf{c}' - \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A} \\ \mathbf{B}^{-1}\mathbf{b} & \mathbf{B}^{-1}\mathbf{A} \end{bmatrix}.$$

To see this is true, note that
$$\begin{aligned} \mathbf{Q}\mathbf{B}^{-1}\overline{\mathbf{B}} &= \mathbf{Q}\mathbf{B}^{-1} \begin{bmatrix} \mathbf{A}_{B(1)} & \dots & \mathbf{A}_{B(l-1)} & \mathbf{A}_j & \mathbf{A}_{B(l+1)} & \dots & \mathbf{A}_{B(m)} \end{bmatrix} \\ &= \mathbf{Q} \begin{bmatrix} \mathbf{e}_1 & \dots & \mathbf{e}_{l-1} & \mathbf{B}^{-1}\mathbf{A}_j & \mathbf{e}_{l+1} & \dots & \mathbf{e}_m \end{bmatrix} \\ &= \mathbf{I}. \end{aligned}$$

Therefore, $\mathbf{Q}\mathbf{B}^{-1} = \overline{\mathbf{B}}^{-1}$. This establishes equations from rows 1 to $m$.

To see the zeroth row, it is sufficient to show that $\mathbf{c}_B'\mathbf{B}^{-1} - k\mathbf{e}_l'\mathbf{B}^{-1} = \mathbf{c}_{\overline{B}}'\overline{\mathbf{B}}^{-1}$. Let $\mathbf{p}' = \mathbf{c}_B'\mathbf{B}^{-1} - k\mathbf{e}_l'\mathbf{B}^{-1}$. For the pivoting column $j$, we clearly have $c_j - \mathbf{p}'\mathbf{A}_j = c_j - \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A}_j + k\mathbf{e}_l'\mathbf{B}^{-1}\mathbf{A}_j = 0$. For basic columns $B(i), i \neq l$, we have $c_{B(i)} - \mathbf{p}'\mathbf{A}_{B(i)} = c_{B(i)} - \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A}_{B(i)} + k\mathbf{e}_l'\mathbf{B}^{-1}\mathbf{A}_{B(i)} = \bar{c}_{B(i)} + k\mathbf{e}_l'\mathbf{e}_i = 0$. That is,
$$\mathbf{c}_{\overline{B}}' - \mathbf{p}'\overline{\mathbf{B}} = \mathbf{0}',$$
and we indeed have $\mathbf{p}' = \mathbf{c}_B'\mathbf{B}^{-1} - k\mathbf{e}_l'\mathbf{B}^{-1} = \mathbf{c}_{\overline{B}}'\overline{\mathbf{B}}^{-1}$.

**Example 3.8** *Consider the linear programming problem*

$$\begin{array}{ll} \min & -10x_1 - 12x_2 - 12x_3 \\ & \begin{array}{llll} x_1+ & x_2+ & 2x_3 & \leq 20 \\ 2x_1+ & x_2+ & 2x_3 & \leq 20 \\ 2x_1+ & 2x_2+ & x_3 & \leq 20 \end{array} \\ s.t. & \\ & x_1, x_2, x_3 \geq 0. \end{array}$$

*The problem can be reformulated into the standard form by introducing slack variables* $x_4, x_5, x_6$*:*

$$\begin{array}{ll} \min & -10x_1 - 12x_2 - 12x_3 \\ & \begin{array}{lllllll} x_1+ & x_2+ & 2x_3+ & x_4 & & & = 20 \\ 2x_1+ & x_2+ & 2x_3+ & & x_5 & & = 20 \\ 2x_1+ & 2x_2+ & x_3+ & & & x_6 & = 20 \end{array} \\ s.t. & \\ & x_1, ..., x_6 \geq 0. \end{array}$$

*Clearly,* $\mathbf{x} = (0, 0, 0, 20, 20, 20)$ *is a basic feasible solution. The corresponding basis matrix is*

$$\mathbf{B} = \mathbf{B}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

*and* $\mathbf{c}_B = \mathbf{0}$*. Thus,* $\mathbf{c}'_B \mathbf{x}_B = 0$ *and* $\mathbf{c}' - \mathbf{c}'_B \mathbf{B}^{-1} \mathbf{A} = \mathbf{c}'$*. The initial tableau is: Since the reduced cost of* $x_1$

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|
| 0 | $-10$ | $-12$ | $-12$ | 0 | 0 | 0 |
| 20 | 1 | 2 | 2 | 1 | 0 | 0 |
| 20 | $2^*$ | 1 | 2 | 0 | 1 | 0 |
| 20 | 2 | 2 | 1 | 0 | 0 | 1 |

*is negative, we let it enter the basis. The pivot column is* $(u_1, u_2, u_3) = (1, 2, 2)$ *and the ratios* $x_i/u_i$ *can be computed easily. We find that the smallest ratio is* 10 *and we can choose* $l = 2$ *(corresponds to* $x_5$*). The* pivot element*, the intersection of pivot column and pivot row, is indicated by asterisk. The next step is to change the pivot column to the vector* $(0, 0, 1, 0)$*, which is done below:*

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|
| 100 | 0 | $-7$ | $-2$ | 0 | 5 | 0 |
| 10 | 0 | 1.5 | $1^*$ | 1 | $-0.5$ | 0 |
| 10 | 1 | 0.5 | 1 | 0 | 0.5 | 0 |
| 0 | 0 | 1 | $-1$ | 0 | $-1$ | 1 |

*We have moved to the new basic feasible solution:* $\mathbf{x} = (10, 0, 0, 10, 0, 0)$*. Again, the reduced cost of, say,* $x_3$ *is negative, and by computing the ratio, we can determine the first row as the pivot row. Continuing iterations, we get Since all the reduced costs in the last tableau are now nonnegative, we have arrived at an optimal solution which is* $\mathbf{x} = (4, 4, 4, 0, 0, 0)$*.*

## 3.4 Finding Initial BFS

The task is easy when

$$P = \{\mathbf{x} | \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}.$$

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|
| 120 | 0 | −4 | 0 | 2 | 4 | 0 |
| 10 | 0 | 1.5 | 1 | 1 | −0.5 | 0 |
| 0 | 1 | −1 | 0 | −1 | 1 | 0 |
| 10 | 0 | 2.5* | 0 | 1 | −1.5 | 1 |

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|
| 136 | 0 | 0 | 0 | 3.6 | 1.6 | 1.6 |
| 4 | 0 | 0 | 1 | 0.4 | 0.4 | −0.6 |
| 4 | 1 | 0 | 0 | −0.6 | 0.4 | 0.4 |
| 4 | 0 | 1 | 0 | 0.4 | −0.6 | 0.4 |

For standard form problem

$$\begin{aligned} \min \quad & \mathbf{c}'\mathbf{x} \\ s.t. \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

one can use the following so-called *big-M method*. We assume without loss of generality that $\mathbf{b} \geq \mathbf{0}$. In the big-M method, we solve instead for the problem

$$\begin{aligned} \min \quad & \mathbf{c}'\mathbf{x} + M \sum_{i=1}^{m} y_i \\ s.t. \quad & \mathbf{A}\mathbf{x} + \mathbf{y} = \mathbf{b}, \\ & \mathbf{x}, \mathbf{y} \geq \mathbf{0}, \end{aligned}$$

where $M$ is a large positive number. In this case, the solution

$$\mathbf{x} = \mathbf{0}, \mathbf{y} = \mathbf{b}$$

is a basic feasible solution. If the original problem has finite optimal cost, then the optimal solution to the modified problem $(\mathbf{x}^*, \mathbf{y}^*)$ must satisfy $\mathbf{y}^* = \mathbf{0}$ and $\mathbf{x}^*$ is the optimal solution to the original problem.

# References

[BT97] D. Bertsimas and J.N. Tsitsiklis, *Introduction to Linear Optimization*, Springer, 1997.

**BDC6111: Introduction to Optimization** **Fall 2018, NUS**

# Lecture 4: Duality Theory

*Lecturer: Zhenyu Hu*

## 4.1 Method of Lagrangian Relaxation

### Motivating Example

The duality theory is intimately related with the Lagrangian relaxation method, which is commonly used to solve difficult constrained optimization problems. To motivate, let's consider the following nonlinear optimization problem:

$$\begin{aligned} \min \quad & x^2 + y^2 \\ s.t. \quad & x + y = 1. \end{aligned} \tag{4.1}$$

One common way of solving such constrained problems is to relax the constraints and consider the following unconstrained problem

$$\min \quad x^2 + y^2 + p(1 - x - y), \tag{4.2}$$

where $p$ is an arbitrary constant. We call $p$ the *Lagrangian multiplier* and the function $L(x, y, p) = x^2 + y^2 + p(1 - x - y)$ the *Lagrangian.*

We make following observations regarding the two problems:

- For any $p$, the optimal value of problem (4.2) provides a lower bound to the optimal value of problem (4.1);
- If $p > 0$, it penalizes the case when $x + y < 1$; If $p < 0$, it penalizes the case when $x + y > 1$.

For this particular example, we can solve the unconstrained problem easily by first order conditions:

$$\frac{\partial L}{\partial x} = 2x - p = 0, \quad \frac{\partial L}{\partial y} = 2y - p = 0,$$

which results in $x = y = \frac{p}{2}$ and an optimal value

$$g(p) = p - \frac{p^2}{2}.$$

Since for each $p$, $g(p)$ is a lower abound on the optimal value of problem (4.1), we want this lower bound a be as tight as possible. This leads to the following problem:

$$\max \quad g(p),$$

whose solution is $p^* = 1$. The corresponding solution to the relaxed problem (4.2) when $p = 1$ is then $x^* = y^* = \frac{1}{2}$. This is indeed a feasible and optimal solution to the original constrained problem (4.1).

## Standard Form Linear Programming Problems

Let us apply this idea to the linear programming problem in standard form

$$\begin{aligned} \min \quad & \mathbf{c}'\mathbf{x} \\ s.t. \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

which we call *primal* problem. We assume that the primal admits a solution $\mathbf{x}^*$ and consequently its optimal value is $\mathbf{c}'\mathbf{x}^*$. The relaxed problem is then defined by replacing the constraint $\mathbf{A}\mathbf{x} = \mathbf{b}$ with the penalty $\mathbf{p}'(\mathbf{b} - \mathbf{A}\mathbf{x})$, i.e., we associate a penalty or Lagrangian multiplier $p_i$ to each constraint $\mathbf{a}_i'\mathbf{x} = b_i$, $i = 1, ..., m$:

$$\begin{aligned} g(\mathbf{p}) = \min \quad & \mathbf{c}'\mathbf{x} + \mathbf{p}'(\mathbf{b} - \mathbf{A}\mathbf{x}) \\ s.t. \quad & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Again, we have $g(\mathbf{p}) \leq \mathbf{c}'\mathbf{x}^*$ and we want to find a penalty such that this lower bound is as tight as possible:

$$\max \quad g(\mathbf{p}).$$

We call this problem as the *dual* problem.

The relaxed problem can be solved easily. We can rewrite $g(\mathbf{p})$ as

$$g(\mathbf{p}) = \mathbf{p}'\mathbf{b} + \min_{\mathbf{x} \geq \mathbf{0}} \quad (\mathbf{c}' - \mathbf{p}\mathbf{A})\mathbf{x}.$$

Note that

$$\min_{\mathbf{x} \geq \mathbf{0}} \quad (\mathbf{c}' - \mathbf{p}\mathbf{A})\mathbf{x} = \begin{cases} 0, & \text{if} \quad \mathbf{c}' - \mathbf{p}'\mathbf{A} \geq \mathbf{0}', \\ -\infty, & \text{otherwise.} \end{cases}$$

If we want to maximize $g(\mathbf{p})$, then we want to enforce $\mathbf{c}' - \mathbf{p}'\mathbf{A} \geq \mathbf{0}'$. The dual problem can then be expressed as

$$\begin{aligned} \max \quad & \mathbf{p}'\mathbf{b} \\ s.t. \quad & \mathbf{p}'\mathbf{A} \leq \mathbf{c}', \end{aligned}$$

which is another linear programming problem with decision variables $\mathbf{p} = (p_1, ..., p_m)$.

## Variants

1. What if we have inequality constraints $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ instead of equality constraints $\mathbf{A}\mathbf{x} = \mathbf{b}$? Intuitively speaking, in the case of inequality constraints, we only want to put a penalty $p_i(b_i - \mathbf{a}_i'\mathbf{x})$ when $\mathbf{a}_i'\mathbf{x} < b_i$ for some $i$. This is achieved by restricting $p_i \geq 0$, since $p_i(b_i - \mathbf{a}_i'\mathbf{x}) \geq 0$ when $\mathbf{a}_i'\mathbf{x} < b_i$ (a penalty cost) while $p_i(b_i - \mathbf{a}_i'\mathbf{x}) \leq 0$ when $\mathbf{a}_i'\mathbf{x} \geq b_i$ (no penalty, even reward).

   Equivalently, we can transform the problem into standard form: $\mathbf{A}\mathbf{x} - \mathbf{s} = \mathbf{b}, \mathbf{s} \geq \mathbf{0}$ or

   $$\begin{bmatrix} \mathbf{A} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{s} \end{bmatrix} = \mathbf{b}.$$

   The dual constraints are then

   $$\mathbf{p}' \begin{bmatrix} \mathbf{A} & -\mathbf{I} \end{bmatrix} \leq \begin{bmatrix} \mathbf{c}' & \mathbf{0}' \end{bmatrix},$$

   or simply

   $$\mathbf{p}'\mathbf{A} \leq \mathbf{c}', \quad \mathbf{p} \geq \mathbf{0}.$$

2. What if $\mathbf{x}$ is unconstrained? The relaxed problem becomes

$$g(\mathbf{p}) = \mathbf{p}'\mathbf{b} + \min_{\mathbf{x}} \quad (\mathbf{c}' - \mathbf{p}\mathbf{A})\mathbf{x},$$

   where

$$\min_{\mathbf{x}} \quad (\mathbf{c}' - \mathbf{p}\mathbf{A})\mathbf{x} = \begin{cases} 0, & \text{if} \quad \mathbf{c}' - \mathbf{p}'\mathbf{A} = \mathbf{0}', \\ -\infty, & \text{otherwise.} \end{cases}$$

   Thus, we end up with the constraint $\mathbf{p}'\mathbf{A} = \mathbf{c}'$ in the dual problem.

In general, if we have the primal of the following form,

$$\begin{aligned} \min \quad & \mathbf{c}'\mathbf{x} \\ s.t. \quad & \mathbf{a}_i'\mathbf{x} \geq b_i, \quad i \in M_1 \\ & \mathbf{a}_i'\mathbf{x} \leq b_i, \quad i \in M_2, \\ & \mathbf{a}_i'\mathbf{x} = b_i, \quad i \in M_3, \\ & x_j \geq 0, \quad j \in N_1, \\ & x_j \leq 0, \quad j \in N_2, \\ & x_j \quad \text{free}, \quad j \in N_3, \end{aligned}$$

the dual is then given by

$$\begin{aligned} \max \quad & \mathbf{p}'\mathbf{b} \\ s.t. \quad & p_i \geq 0, \quad i \in M_1, \\ & p_i \leq 0, \quad i \in M_2, \\ & p_i \quad \text{free}, \quad i \in M_3, \\ & \mathbf{p}'\mathbf{A}_j \leq c_j, \quad j \in N_1 \\ & \mathbf{p}'\mathbf{A}_j \geq c_j, \quad j \in N_2 \\ & \mathbf{p}'\mathbf{A}_j = c_j, \quad j \in N_3 \end{aligned}$$

**Remark 4.1** *We observe here that the quantities defined by $u_i = p_i(\mathbf{a}_i'\mathbf{x} - b_i)$ and $v_j = (c_j - \mathbf{p}'\mathbf{A}_j)x_j$ are always nonnegative, i.e., $u_i, v_j \geq 0$ for all $1 \leq i \leq m$ and $1 \leq j \leq n$.*

**Example 4.2** *Consider the primal problem*

$$\begin{array}{ll} \min & x_1 + 2x_2 + 3x_3 \\ & \begin{array}{rrrl} -x_1+ & 3x_2 & & = 5 \\ 2x_1- & x_2+ & 3x_3 & \geq 6 \\ & & x_3 & \leq 4 \end{array} \\ s.t. & \\ & x_1 \geq 0 \\ & x_2 \leq 0 \\ & x_3 \quad \textit{free} \end{array}$$

*Its dual is*

$$
\begin{array}{rl}
\max & 5p_1 + 6p_2 + 4p_3 \\
s.t. & p_1 \quad \textit{free} \\
& p_2 \geq 0 \\
& p_3 \leq 0 \\
& \begin{array}{rrrl}
-p_1 + & 2p_2 + & & \leq 1 \\
3p_1 - & p_2 + & & \geq 2 \\
& 3p_2 + & p_3 & = 3
\end{array}
\end{array}
$$

*What is the dual of the dual? We first transform it into the form of primal:*

$$
\begin{array}{rl}
-\min & -5x_1 - 6x_2 - 4x_3 \\
s.t. & x_1 \quad \textit{free} \\
& x_2 \geq 0 \\
& x_3 \leq 0 \\
& \begin{array}{rrrl}
x_1 - & 2x_2 + & & \geq -1 \\
-3x_1 + & x_2 + & & \leq -2 \\
& -3x_2 & -x_3 & = -3
\end{array}
\end{array}
$$

*The dual of this problem can then be found according to our general formulation as*

$$
\begin{array}{rl}
-\max & -p_1 - 2p_2 - 3p_3 \\
s.t. & \begin{array}{rrrl}
p_1 - & 3p_2 + & & = -5 \\
2p_1 + & p_2 - & 3p_3 & \leq -6 \\
& & -p_3 & \geq -4
\end{array} \\
& p_1 \geq 0 \\
& p_2 \leq 0 \\
& p_3 \quad \textit{free}
\end{array}
$$

*This is the same problem as the primal we started with. We usually state this property as* the dual of the dual is the primal.

## 4.2 Duality Theory

We have mentioned above for the standard form linear programming problems, if the optimal solution to the primal $\mathbf{x}^*$ exists, then $g(\mathbf{p}) \leq \mathbf{c}'\mathbf{x}^*$. This is true in general.

**Theorem 4.3 (Weak Duality)** *If $\mathbf{x}$ is a feasible solution to the primal problem and $\mathbf{p}$ is a feasible solution to the dual problem, then*

$$\mathbf{p}'\mathbf{b} \leq \mathbf{c}'\mathbf{x}.$$

*In particular, when $\mathbf{x}^*$ exists this implies $g(\mathbf{p}) \leq \mathbf{c}'\mathbf{x}^*$.*

*Proof:* Recall the quantities $u_i, v_j$. Notice that

$$\sum_{i=1}^{m} u_i = \mathbf{p}'(\mathbf{A}\mathbf{x} - \mathbf{b}) \geq 0$$

and

$$\sum_{j=1}^{n} v_j = (\mathbf{c}' - \mathbf{p}'\mathbf{A})\mathbf{x} \geq 0.$$

We then have

$$\sum_{i=1}^{m} u_i + \sum_{j=1}^{n} v_j = \mathbf{c}'\mathbf{x} - \mathbf{p}'\mathbf{b} \geq 0.$$

That is,

$$\mathbf{p}'\mathbf{b} \leq \mathbf{c}'\mathbf{x}.$$

■

Weak duality theorem immediately implies:

- If the optimal cost in the primal is $-\infty$, then the dual problem must be infeasible;
- If the optimal cost in the dual is $+\infty$, then the primal problem must be infeasible;
- Let $\mathbf{x}$ and $\mathbf{p}$ be the feasible solutions to the primal and the dual. If $\mathbf{p}'\mathbf{b} = \mathbf{c}'\mathbf{x}$, then $\mathbf{x}$ and $\mathbf{p}$ are optimal solutions to the primal and the dual, respectively.

**Theorem 4.4 (Strong Duality)** *If a linear programming problem has an optimal solution, then its dual also has a solution and the respective optimal costs are equal.*

*Proof:* Consider the standard form problem

$$\begin{aligned} \min \quad & \mathbf{c}'\mathbf{x} \\ s.t. \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

For simplicity, we assume the rows of $\mathbf{A}$ are linear independent and there exists an optimal solution. Let us apply the simplex method to this problem. We have shown that when the basic feasible solutions are non-degenerate then the simplex method terminates. (In the case when there are degenerate basic feasible solutions, by using specific pivoting rule the simplex method can still terminate.) Let the simplex method terminates at the solution $\mathbf{x}$ and the corresponding basis is $\mathbf{B}$. Recall that the basic variables are given by $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$ and the reduced costs at the solution must have

$$\mathbf{c}' - \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A} \geq \mathbf{0}'.$$

where $\mathbf{c}_B'$ is the costs associated with basic variables.

Let $\mathbf{p}' = \mathbf{c}_B'\mathbf{B}^{-1}$, we then have

$$\mathbf{p}'\mathbf{A} \leq \mathbf{c}'.$$

This implies $\mathbf{p}$ is a dual feasible solution. In addition,

$$\mathbf{p}'\mathbf{b} = \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{b} = \mathbf{c}_B'\mathbf{x}_B = \mathbf{c}\mathbf{x},$$

which proves that $\mathbf{p}$ is an optimal solution to the dual and the optimal values of the primal and dual are equal. ■

**Remark 4.5** *The proof shows that when an optimal solution to the primal is obtained by simplex method, the optimal solution to the dual is also obtained as* $\mathbf{p}' = \mathbf{c}'_B\mathbf{B}^{-1}$ *and we do not need to re-solve the dual program.*

**Remark 4.6** *The proof hinges on the fact that when the simplex algorithm terminates, the reduced costs must be nonnegative—this is clearly guaranteed if the optimal solution is nondegenerate. However, when the optimal solution is degenerate, it is possible that $\bar{c}_j < 0$ for some nonbasic index $j$. The pivoting rules (see Section 3.4 in [BT97]) that deals with degeneracy issue guarantees that the simplex algorithm terminates in finite iterations with all reduced costs at optimal solution being nonnegative.*

We summarize the relation between the optimal values of the primal and dual in the following table. The

| | **Finite optimum** | **Unbounded** | **Infeasible** |
|---|---|---|---|
| **Finite optimum** | Possible | Impossible | Impossible |
| **Unbounded** | Impossible | Impossible | Possible |
| **Infeasible** | Impossible | Possible | Possible |

table is explained as follows:

- When the primal has finite optimum, by strong duality, the dual also has one.
- When the primal is unbounded, by weak duality, the dual has to be infeasible.
- When the primal is infeasible, on the other hand, it is possible that the dual is also infeasible.

**Example 4.7** *Consider the primal*

$$\begin{array}{ll} \min & x_1 + 2x_2 \\ s.t. & \begin{array}{rrl} x_1+ & x_2 & = 1 \\ 2x_1+ & 2x_2 & = 3 \end{array} \end{array}$$

*Its dual is*

$$\begin{array}{ll} \max & p_1 + 3p_2 \\ s.t. & \begin{array}{rrl} p_1+ & 2p_2 & = 1 \\ p_1+ & 2p_2 & = 2 \end{array} \end{array}$$

*Both problems are infeasible.*

**Theorem 4.8 (Complementary Slackness)** *Let $\mathbf{x}$ and $\mathbf{p}$ be feasible solutions to the primal and the dual problem, respectively. The vectors $\mathbf{x}$ and $\mathbf{p}$ are optimal solutions for the two respective problems if and only if*

$$u_i = p_i(\mathbf{a}_i'\mathbf{x} - b_i) = 0$$

*and*

$$v_j = (c_j - \mathbf{p}'\mathbf{A}_j)x_j = 0$$

*for all $1 \leq i \leq m$ and $1 \leq j \leq n$.*

*Proof:* Recall in the proof of Theorem 4.3 that

$$\sum_{i=1}^{m} u_i = \mathbf{p}'(\mathbf{A}\mathbf{x} - \mathbf{b}) \geq 0$$

and

$$\sum_{j=1}^{n} v_j = (\mathbf{c}' - \mathbf{p}'\mathbf{A})\mathbf{x} \geq 0.$$

Thus,

$$\sum_{i=1}^{m} u_i + \sum_{j=1}^{n} v_j = \mathbf{c}'\mathbf{x} - \mathbf{p}'\mathbf{b} \geq 0.$$

If $\mathbf{x}$ and $\mathbf{p}$ are primal and dual optimal, by strong duality $\mathbf{c}'\mathbf{x} - \mathbf{p}'\mathbf{b} = 0$ which implies $\sum_{i=1}^{m} u_i + \sum_{j=1}^{n} v_j$. Since $u_i, v_j \geq 0$, we must then have $u_i = v_j = 0$.

On the other hand, if $u_i = v_j = 0$, then $\mathbf{c}'\mathbf{x} - \mathbf{p}'\mathbf{b} = 0$, which implies $\mathbf{x}$ and $\mathbf{p}$ are primal and dual optimal. ■

**Example 4.9** *Consider the primal*

$$\begin{array}{ll} \min & 3x_1 + x_2 \\ s.t. & \begin{array}{lllll} x_1 + & x_2 - & x_3 & & = 2 \\ 2x_1 - & x_2 & & -x_4 & = 0 \end{array} \\ & x_1, x_2, x_3, x_4 \geq 0 \end{array}$$

*and its dual*

$$\begin{array}{ll} \max & 2p_1 \\ s.t. & \begin{array}{lll} p_1 + & 2p_2 & \leq 3 \\ p_1 - & p_2 & \leq 1 \\ -p_1 & & \leq 0 \\ & -p_2 & \leq 0. \end{array} \end{array}$$

*Consider the nondegenerate optimal solution* $\mathbf{x}^* = (2/3, 4/3, 0, 0)$. *What is the optimal solution to the dual problem?*

One can see that when $\mathbf{x}^*$ is nondegenerate, we have $x^*_{B(i)} > 0, i = 1, ..., m$. The complementary slackness conditions then imply

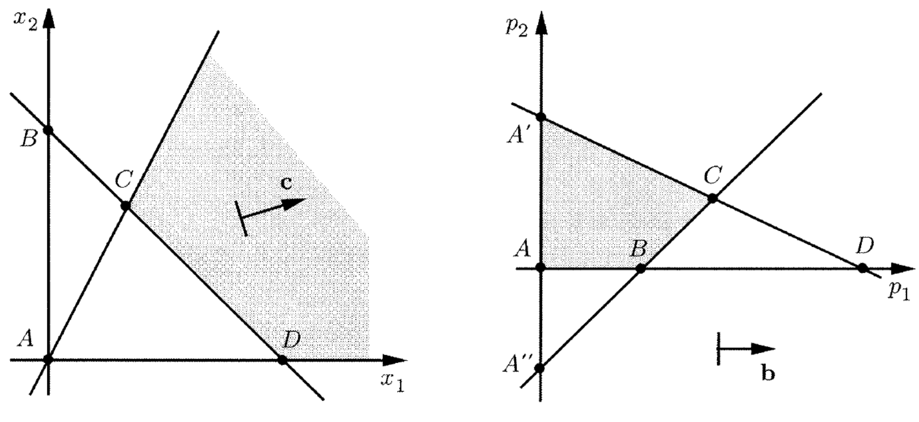
$$c_{B(i)} - \mathbf{p}'\mathbf{A}_{B(i)} = 0, i = 1, ..., m$$

or equivalently $\mathbf{c}'_B - \mathbf{p}'\mathbf{B} = 0$, which has a unique solution $\mathbf{p}' = \mathbf{c}'_B\mathbf{B}^{-1}$. In fact, for any basic feasible solution $\mathbf{x}$ to the primal problem with $\mathbf{B}$ being its basis, the vector $\mathbf{p}' = \mathbf{c}'_B\mathbf{B}^{-1}$ specifies a basic solution (not necessarily feasible) to the dual problem, which always satisfies $\mathbf{p}'\mathbf{b} = \mathbf{c}'_B\mathbf{B}^{-1}\mathbf{b} = \mathbf{c}'_B\mathbf{x}_B = \mathbf{c}'\mathbf{x}$. It follows that when $\mathbf{p}'$ becomes feasible, then both $\mathbf{p}'$ and the corresponding basic feasible solution $\mathbf{x}$ become optimal. In this sense, *reduced cost can also be interpreted as the extend to which the dual basic solution violates each constraint.* The figure below illustrates the basic solutions of the primal and the corresponding basic solutions of the dual for Example 4.9.

## 4.3 Geometric Interpretation of Duality and Farkas' Lemma

For our illustration, we use the following example with the feasible set in the form of $\mathbf{A}\mathbf{x} \geq \mathbf{b}$:

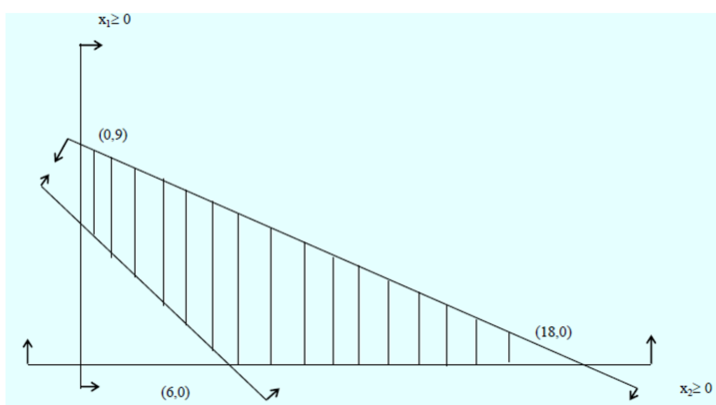
$$\begin{array}{ll} \min & 2x_1 + 5x_2 \\ s.t. & x_1 + x_2 \geq 6, \\ & -x_1 - 2x_2 \geq -18, \\ & x_1 \geq 0, \\ & x_2 \geq 0. \end{array}$$




The dual of this problem is given by

$$
\begin{aligned}
\max \quad & 6p_1 - 18p_2 \\
s.t. \quad & p_1 - p_2 + p_3 = 2, \\
& p_1 - 2p_2 + p_4 = 5, \\
& p_1, p_2, p_3, p_4 \geq 0.
\end{aligned}
$$

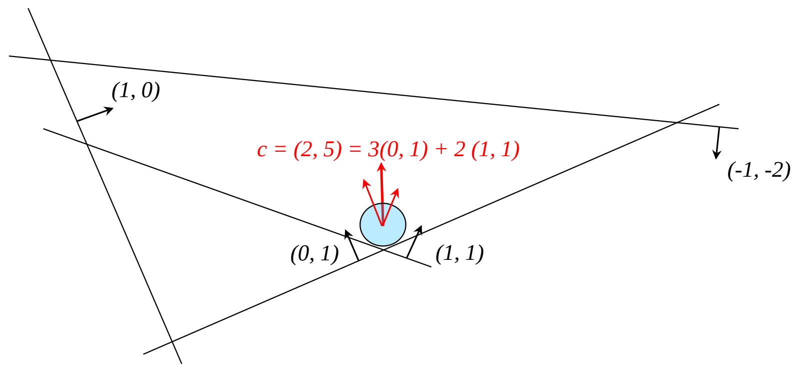
The feasible set of the primal problem is given in the figure below.




Without any constraints, the objective in the primal will decrease in a fastest way along the direction $(-2, -5)$. We can think of this direction as a direction of gravity and we can rotate the picture accordingly as illustrated in the following figure. One can see that under the gravity, the ball will fall to the point $(6, 0)$ which is the primal optimal solution. What is the dual optimal solution? The ball stops at $(6, 0)$ and its gravity has to be counterbalanced by the forces from the "walls", which have directions $(0, 1)$ and $(1, 1)$ respectively. In other words, $\mathbf{c}$ can be represented as a positive combination of $(0, 1)$ and $(1, 1)$ at the optimal point. Indeed, we have

$$\mathbf{c} = 3(0, 1) + 2(1, 1).$$

The weight $(3, 2)$ corresponds to the dual optimal solution $(p_1^*, p_2^*, p_3^*, p_4^*) = (2, 0, 0, 3)$. One can also see that the "walls": $-x_1 - 2x_2 \geq -18$ and $x_1 = 0$ can be removed without affecting the position of the ball. The complementary slackness conditions then imply $p_2^* = p_3^* = 0$.




The underlying general geometric result is called Farkas's lemma:

**Theorem 4.10 (Farkas' Lemma)** *Let $\mathbf{a}_1, ..., \mathbf{a}_m$ and $\mathbf{c}$ be vectors in $\mathbb{R}^n$. Then, exactly one of the following alternatives holds:*

1. *There exists some $\mathbf{p} \geq \mathbf{0}$ in $\mathbb{R}^m$ such that*

$$\sum_{i=1}^{m} p_i \mathbf{a}_i = \mathbf{c}.$$

2. *There exists some vector $\mathbf{d} \in \mathbb{R}^n$, such that*

$$\mathbf{Ad} = \begin{bmatrix} \mathbf{a}_1' \\ \vdots \\ \mathbf{a}_m' \end{bmatrix} \mathbf{d} \geq \mathbf{0}$$

*and $\mathbf{c}'\mathbf{d} < 0$.*

*Proof*: Suppose the first alternative holds, and consider any vector $\mathbf{d} \in \mathbb{R}^n$, such that

$$\mathbf{Ad} = \begin{bmatrix} \mathbf{a}_1' \\ \vdots \\ \mathbf{a}_m' \end{bmatrix} \mathbf{d} \geq \mathbf{0}.$$

Since $\mathbf{p} \geq \mathbf{0}$, we must have $\mathbf{p}'\mathbf{Ad} \geq 0$. It then follows that

$$\mathbf{p}'\mathbf{Ad} = \mathbf{c}'\mathbf{d} \geq 0,$$

and the second alternative can never hold.

Now suppose the first alternative does not hold, i.e., we cannot find a vector $\mathbf{p} \geq \mathbf{0}$ in $\mathbb{R}^m$ such that

$$\sum_{i=1}^{m} p_i \mathbf{a}_i = \mathbf{c}.$$

Consider the following problem

$$\begin{aligned} \min \quad & \mathbf{c}'\mathbf{x} \\ s.t. \quad & \mathbf{A}\mathbf{x} \geq \mathbf{0} \end{aligned}$$

and its dual

$$\begin{aligned} \max \quad & \mathbf{p}'\mathbf{0} \\ s.t. \quad & \mathbf{p}'\mathbf{A} = \mathbf{c}' \\ & \mathbf{p} \geq \mathbf{0}. \end{aligned}$$

The invalidity of the first alternative implies the dual problem is infeasible. By duality theory, we know that the primal problem is either infeasible or has unbounded cost. Clearly, $\mathbf{x} = \mathbf{0}$ is a feasible solution. Hence, there must exist some vector $\mathbf{d} \in \mathbb{R}^n$, such that $\mathbf{A}\mathbf{d} \geq \mathbf{0}$ and $\mathbf{c}'\mathbf{d} < 0$. ■

Using the ball example, Farkas' Lemma can be interpreted as either the ball reaches to a equilibrium where all forces are balanced (the first statement) or there is a feasible direction $\mathbf{d}$ such that the ball can further fall (cost can be reduced).

As another geometric interpretation of Farkas' Lemma, it states that either the vector $\mathbf{c}$ can be expressed as a nonnegative combination of the vectors $\mathbf{a}_1, ..., \mathbf{a}_m$ or there must exist a vector $\mathbf{d}$ such that the hyperplane $\{\mathbf{x} | \mathbf{d}'\mathbf{x} = 0\}$ separates the vector $\mathbf{c}$ and the vectors $\mathbf{a}_1, ..., \mathbf{a}_m$.

# References

[BT97] D. Bertsimas and J.N. Tsitsiklis, *Introduction to Linear Optimization*, Springer, 1997.

| | |
|---|---|
| **BDC6111: Introduction to Optimization** | **Fall 2018, NUS** |

# Lecture 5: Sensitivity Analysis

*Lecturer: Zhenyu Hu*

Consider the standard form problem

$$\begin{aligned} \min \quad & \mathbf{c}'\mathbf{x} \\ s.t. \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

and its dual

$$\begin{aligned} \max \quad & \mathbf{p}'\mathbf{b} \\ s.t. \quad & \mathbf{p}'\mathbf{A} \leq \mathbf{c}'. \end{aligned}$$

Suppose the rows of $\mathbf{A}$ are linearly independent and we already have an optimal basis $\mathbf{B}$ and the associated optimal solution $\mathbf{x}^*$.

Since in practice we often have incomplete knowledge of problem data, we wish to understand when certain problem parameters are changed

- how the optimal cost would change;
- whether the current basis is still optimal.

## 5.1 Local Sensitivity Analysis

### Adding a new variable

Suppose that we introduce a new variable $x_{n+1}$, together with a cost coefficient $c_{n+1}$ and corresponding column $\mathbf{A}_{n+1}$, and obtain the new problem

$$\begin{aligned} \min \quad & \mathbf{c}'\mathbf{x} + c_{n+1}x_{n+1} \\ s.t. \quad & \mathbf{A}\mathbf{x} + \mathbf{A}_{n+1}x_{n+1} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, x_{n+1} \geq 0. \end{aligned}$$

- Since $(\mathbf{x}, x_{n+1}) = (\mathbf{x}^*, 0)$ is a basic feasible solution to the new problem, it is guaranteed that the optimal cost will not increase.
- At basis $\mathbf{B}$, $x_{n+1}$ is a nonbasic variable and has reduced cost

  $$\bar{c}_{n+1} = c_{n+1} - \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A}_{n+1}.$$

  If $\bar{c}_{n+1} \geq 0$, then $(\mathbf{x}, x_{n+1}) = (\mathbf{x}^*, 0)$ is an optimal solution to the new problem and the optimal cost will remain unchanged. If, on the other hand, $\bar{c}_{n+1} < 0$, then we can perform a simplex iteration by letting $\mathbf{A}_{n+1}$ into the basis.

**Example 5.1** *Consider the problem*

$$\begin{aligned} \min \quad & -5x_1 - x_2 + 12x_3 \\ s.t. \quad & 3x_1 + 2x_2 + x_3 = 10, \\ & 5x_1 + 3x_2 \qquad + x_4 = 16, \\ & x_1, x_2, x_3, x_4 \geq 0. \end{aligned}$$

*An optimal solution to this problem is given by* $\mathbf{x} = (2, 2, 0, 0)$ *and the corresponding simplex tableau is given by*

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|---|
| 12 | 0 | 0 | 2 | 7 |
| 2 | 1 | 0 | $-3$ | 2 |
| 2 | 0 | 1 | 5 | $-3$ |

*Let us now introduce a variable* $x_5$ *and consider the new problem*

$$\begin{aligned} \min \quad & -5x_1 - x_2 + 12x_3 \qquad - x_5 \\ s.t. \quad & 3x_1 + 2x_2 + x_3 \qquad + x_5 = 10, \\ & 5x_1 + 3x_2 \qquad + x_4 + x_5 = 16, \\ & x_1, x_2, x_3, x_4, x_5 \geq 0, \end{aligned}$$

*where* $\mathbf{A}_5 = (1, 1)$*.*

*To see if the new problem has a lower cost, we compute the reduced cost for* $x_5$*:*

$$\bar{c}_5 = c_5 - \mathbf{c}_B' \mathbf{B}^{-1} \mathbf{A}_5 = -1 - [-5, -1] \begin{bmatrix} -3 & 2 \\ 5 & -3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = -4.$$

*Indeed, using simplex to bring* $x_5$ *into the basis and* $x_2$ *out of the basis, the new solution can be given by* $\mathbf{x} = (3, 0, 0, 0, 1)$*.*

## Adding a new equality constraint

Consider now a new constraint

$$\mathbf{a}_{m+1}' \mathbf{x} = b_{m+1}$$

is added with $\mathbf{a}_{m+1}' = (a_1, ..., a_n)$.

- If $\mathbf{x}^*$ satisfies $\mathbf{a}_{m+1}' \mathbf{x}^* = b_{m+1}$, then $\mathbf{x}^*$ is also optimal to the new problem.
- If $\mathbf{x}^*$ violates $\mathbf{a}_{m+1}' \mathbf{x} = b_{m+1}$, we assume without loss of generality that $\mathbf{a}_{m+1}' \mathbf{x}^* > b_{m+1}$. To obtain a new solution without resolving the problem again, we consider

$$\begin{aligned} \min \quad & \mathbf{c}' \mathbf{x} + M x_{n+1} \\ s.t. \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{a}_{m+1}' \mathbf{x} - x_{n+1} = b_{m+1}, \\ & \mathbf{x} \geq \mathbf{0}, x_{n+1} \geq 0, \end{aligned}$$

where $M$ is a large positive constant. Note that $(\mathbf{x}^*, x_{n+1})$, where $x_{n+1} = \mathbf{a}'_{m+1}\mathbf{x}^* - b_{m+1}$ is a basic feasible solution to this problem with the basis

$$\bar{\mathbf{B}} = \begin{bmatrix} \mathbf{A}_{B(1)} & \dots & \mathbf{A}_{B(m)} & \mathbf{0} \\ a_{B(1)} & \dots & a_{B(m)} & -1 \end{bmatrix}$$

One can then apply simplex method to this problem starting from $(\mathbf{x}^*, x_{n+1})$ without solving again from the scratch.

## Changing the vector b

Suppose that some component $b_i$ of the requirement vector $\mathbf{b}$ is changed to $b_i + \delta$. Alternatively, we can write $\mathbf{b}$ is changed to $\mathbf{b} + \delta\mathbf{e}_i$ and the new problem is:

$$\begin{aligned} \min \quad & \mathbf{c}'\mathbf{x} \\ s.t. \quad & \mathbf{A}\mathbf{x} = \mathbf{b} + \delta\mathbf{e}_i, \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

In this case, the original solution $\mathbf{x}^*$ clearly becomes infeasible. Yet, changes in $\mathbf{b}$ does not affect the basis $\mathbf{B}$ and the new basic solution corresponding to $\mathbf{B}$ is

$$\mathbf{x}_B = \mathbf{B}^{-1}(\mathbf{b} + \delta\mathbf{e}_i).$$

- By our construction, the equality constraints are clearly satisfied:

$$\sum_{j=1}^{n} \mathbf{A}_j x_j = \mathbf{B}\mathbf{x}_B = \mathbf{B}\mathbf{B}^{-1}(\mathbf{b} + \delta\mathbf{e}_i) = \mathbf{b} + \delta\mathbf{e}_i.$$

- The reduced cost for the new solution is still

$$\mathbf{c}' - \mathbf{c}'_B\mathbf{B}^{-1}\mathbf{A} \geq \mathbf{0}'.$$

- Non-negativity constraints

$$\mathbf{x}_B = \mathbf{B}^{-1}(\mathbf{b} + \delta\mathbf{e}_i) \geq \mathbf{0},$$

  however, may not be satisfied. To see this, let $\mathbf{g} = (\beta_{1i}, \beta_{2i}, ..., \beta_{mi})$ be the ith column of $\mathbf{B}^{-1}$. Then

$$\mathbf{x}_B = \mathbf{x}^*_B + \delta\mathbf{g}.$$

  Thus, for $\mathbf{x}_B \geq \mathbf{0}$, it is sufficient to require

$$x^*_{B(j)} + \delta\beta_{ji} \geq 0, j = 1, ..., m.$$

  This gives a bound on $\delta$:

$$\max_{\{j|\beta_{ji}>0\}} \left(-\frac{x^*_{B(j)}}{\beta_{ji}}\right) \leq \delta \leq \min_{\{j|\beta_{ji}<0\}} \left(-\frac{x^*_{B(j)}}{\beta_{ji}}\right).$$

To sum up, as long as $\delta$ is small enough, the new solution $\mathbf{x}_B = \mathbf{B}^{-1}(\mathbf{b} + \delta\mathbf{e}_i)$ is optimal, and the new optimal cost is

$$\mathbf{c}'_B\mathbf{x}_B = \mathbf{c}'_B\mathbf{B}^{-1}(\mathbf{b} + \delta\mathbf{e}_i).$$

When $\mathbf{x}^*$ is optimal to the original problem, the optimal dual solution for the original problem is

$$\mathbf{p}' = \mathbf{c}_B'\mathbf{B}^{-1}.$$

Thus, the new cost can be further written as

$$\mathbf{c}_B'\mathbf{x}_B = \mathbf{p}'\mathbf{b} + \delta p_i = \underbrace{\mathbf{c}_B'\mathbf{x}_B^*}_{\text{original cost}} + \underbrace{\delta p_i}_{\text{cost variation}},$$

which leads to the following observations:

- When $p_i > 0$, the cost increases;
- When $p_i < 0$, the cost decreases.

## 5.2 Global Dependence

### Dependence on b

Let

$$P(\mathbf{b}) = \{\mathbf{x}|\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$$

and

$$S = \{\mathbf{b}|P(\mathbf{b}) \text{ is nonempty}\}.$$

Note that $S = \{\mathbf{A}\mathbf{x} : \mathbf{x} \geq \mathbf{0}\}$ and is a convex set. For any $\mathbf{b} \in S$, let

$$F(\mathbf{b}) = \min_{\mathbf{x} \in P(\mathbf{b})} \mathbf{c}'\mathbf{x}.$$

We assume that $\{\mathbf{p}|\mathbf{p}'\mathbf{A} \leq \mathbf{c}'\} \neq \emptyset$ so that $F(\mathbf{b})$ is finite for any $\mathbf{b} \in S$.

Suppose for some $\mathbf{b}^* \in S$, there exists a nondegenerate optimal BFS. We let $\mathbf{B}$ be the corresponding basis and by nondegeneracy $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}^* > \mathbf{0}$. As in the local sensitivity analysis, if we change $\mathbf{b}^*$ to $\mathbf{b}$, as long as $\mathbf{b} - \mathbf{b}^*$ is sufficiently small, we have $\mathbf{B}^{-1}\mathbf{b} > \mathbf{0}$ as well and the corresponding optimal cost becomes

$$F(\mathbf{b}) = \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{b} = \mathbf{p}'\mathbf{b}$$

where $\mathbf{p}' = \mathbf{c}_B'\mathbf{B}^{-1}$. That is, $F(\mathbf{b})$ is a linear function in the vicinity of $\mathbf{b}^*$. In general, we have the following result.

**Theorem 5.2** *The optimal cost $F(\mathbf{b})$ is a convex function of $\mathbf{b}$ on the set $S$.*

*Proof:* Let $\mathbf{b}^1, \mathbf{b}^2 \in S$ and $\lambda \in [0, 1]$. We want to show that $F(\lambda\mathbf{b}^1 + (1-\lambda)\mathbf{b}^2) \leq \lambda F(\mathbf{b}^1) + (1-\lambda)F(\mathbf{b}^2)$. Let $\mathbf{x}^1, \mathbf{x}^2$ be the optimal solution corresponding to $\mathbf{b}^1, \mathbf{b}^2$ respectively. Hence, $F(\mathbf{b}^1) = \mathbf{c}'\mathbf{x}^1, F(\mathbf{b}^2) = \mathbf{c}'\mathbf{x}^2$.

Note that $\lambda\mathbf{x}^1 + (1-\lambda)\mathbf{x}^2$ satisfies

$$\mathbf{A}(\lambda\mathbf{x}^1 + (1-\lambda)\mathbf{x}^2) = \lambda\mathbf{b}^1 + (1-\lambda)\mathbf{b}^2$$

and $\lambda\mathbf{x}^1 + (1-\lambda)\mathbf{x}^2 \geq \mathbf{0}$. This implies $\lambda\mathbf{x}^1 + (1-\lambda)\mathbf{x}^2 \in P(\lambda\mathbf{b}^1 + (1-\lambda)\mathbf{b}^2)$. As a result,

$$F(\lambda\mathbf{b}^1 + (1-\lambda)\mathbf{b}^2) \leq \mathbf{c}'(\lambda\mathbf{x}^1 + (1-\lambda)\mathbf{x}^2) = \lambda F(\mathbf{b}^1) + (1-\lambda)F(\mathbf{b}^2).$$

■

By strong duality, we also have

$$\begin{aligned} F(\mathbf{b}) = \max \quad & \mathbf{p}'\mathbf{b} \\ s.t. \quad & \mathbf{p}'\mathbf{A} \leq \mathbf{c}'. \end{aligned}$$

Let $\mathbf{p}^1, ..., \mathbf{p}^N$ be the extreme points of the dual feasible set. It follows that

$$F(\mathbf{b}) = \max_{i=1,\dots,N} (\mathbf{p}^i)'\mathbf{b}, \ \mathbf{b} \in S,$$

which is a piecewise linear convex function.

**Definition 5.3** *Let $F$ be a convex function defined on a convex set $S$. Let $\mathbf{b}^* \in S$. A vector $\mathbf{p}$ is said to be a subgradient of $F$ at $\mathbf{b}^*$ if*

$$F(\mathbf{b}^*) + \mathbf{p}'(\mathbf{b} - \mathbf{b}^*) \leq F(\mathbf{b}), \ \forall \mathbf{b} \in S.$$

When $F(\mathbf{b})$ is differentiable, there is a unique subgradient, equal to the gradient of $F$.

**Theorem 5.4** *Suppose at $\mathbf{b}^* \in S$, the primal problem is feasible and has finite optimal cost. Then, a vector $\mathbf{p}$ is an optimal solution to the dual problem if and only if it is a subgradient of $F$ at the point $\mathbf{b}^*$.*

*Proof:* Suppose $\mathbf{p}$ is an optimal solution to the dual problem. We then have $\mathbf{p}'\mathbf{b}^* = F(\mathbf{b}^*)$. On the other hand, for any $\mathbf{b} \in S$, $\mathbf{p}$ is still feasible and hence we must have $\mathbf{p}'\mathbf{b} \leq F(\mathbf{b})$. Thus,

$$\mathbf{p}'(\mathbf{b} - \mathbf{b}^*) \leq F(\mathbf{b}) - F(\mathbf{b}^*).$$

Now suppose $\mathbf{p}$ is a subgradient of $F$ at $\mathbf{b}^*$. For any $\mathbf{x} \geq \mathbf{0}$, let $\mathbf{b} = \mathbf{A}\mathbf{x}$ such that $\mathbf{b} \in S$. We then have

$$\mathbf{p}'\mathbf{b} = \mathbf{p}'\mathbf{A}\mathbf{x} \leq F(\mathbf{b}) - F(\mathbf{b}^*) + \mathbf{p}'\mathbf{b}^* \leq \mathbf{c}'\mathbf{x} - F(\mathbf{b}^*) + \mathbf{p}'\mathbf{b}^*$$

or equivalently $(\mathbf{c}' - \mathbf{p}'\mathbf{A})\mathbf{x} \geq F(\mathbf{b}^*) - \mathbf{p}'\mathbf{b}^*$ for any $\mathbf{x} \geq \mathbf{0}$. This implies $\mathbf{c}' - \mathbf{p}'\mathbf{A} \geq \mathbf{0}$, i.e., $\mathbf{p}$ is dual feasible. In addition, with $\mathbf{x} = \mathbf{0}$, we have $\mathbf{p}'\mathbf{b}^* \geq F(\mathbf{b}^*)$ and by feasibility $\mathbf{p}'\mathbf{b}^* = F(\mathbf{b}^*)$, i.e., $\mathbf{p}$ is an optimal dual solution. ■

## Dependence on c

We assume the feasible set $\{\mathbf{x} | \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\} \neq \emptyset$. Let

$$Q(\mathbf{c}) = \{\mathbf{p} | \mathbf{p}'\mathbf{A} \leq \mathbf{c}'\}$$

and

$$T = \{\mathbf{c} | Q(\mathbf{c}) \text{ is nonempty}\},$$

which is convex. For any $\mathbf{c} \in T$, let

$$\begin{aligned} G(\mathbf{c}) = \min \quad & \mathbf{c}'\mathbf{x} \\ s.t. \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

then by the definition of $T$, $G(\mathbf{c})$ is finite for any $\mathbf{c} \in T$.

**Theorem 5.5** *The optimal cost $G(\mathbf{c})$ is a piecewise linear concave function of $\mathbf{c}$ on the set $T$. In addition, if for some $\mathbf{c}^* \in T$, the optimal solution has a unique solution $\mathbf{x}$, then $G$ has a gradient of $\mathbf{x}$ at $\mathbf{c}^*$.*

*Proof:* Let $\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^N$ be the extreme points of the feasible set and we have

$$G(\mathbf{c}) = \min_{i=1,\ldots,N} \mathbf{c}'\mathbf{x}^i.$$

Hence, $G(\mathbf{c})$ is a piecewise linear concave function of $\mathbf{c}$.

If for some $\mathbf{c}^* \in T$, the optimal solution has a unique solution $\mathbf{x}^i$, we have

$$(\mathbf{c}^*)'\mathbf{x}^i < (\mathbf{c}^*)'\mathbf{x}^j, \ j \neq i.$$

Then, for $\mathbf{c}$ sufficiently close to $\mathbf{c}^*$, one still has $\mathbf{c}^*\mathbf{x}^i < \mathbf{c}^*\mathbf{x}^j$ for $j \neq i$. In other words, $G(\mathbf{c}) = \mathbf{c}'\mathbf{x}^i$ and hence has gradient $\mathbf{x}^i$ at $\mathbf{c}^*$. ■

# References

[BT97] D. Bertsimas and J.N. Tsitsiklis, *Introduction to Linear Optimization*, Springer, 1997.

**BDC6111: Introduction to Optimization** **Fall 2018, NUS**

# Lecture 6: Applications of Duality Theory

*Lecturer: Zhenyu Hu*

## 6.1 Economic Interpretation of Duality

We interpret duality theory in the context of a production example. Consider a firm producing $n$ products from $m$ resources. Product $j$, $1 \leq j \leq n$ can be sold at a price $c_j$ and requires $a_{ij}$ amount of resource $i$. The firm has $b_i$ amount of resource $i$. If the firm decides to produce $x_j$ units of product $j$, then it must satisfy the resource constraint $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$ and it gives the firm a revenue of $\sum_{j=1}^{n} c_j x_j$. The firm then seeks to solve:

$$\begin{aligned} \max \quad & \mathbf{c}'\mathbf{x} \\ s.t. \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned} \tag{6.1}$$

Now, suppose there is a market (or supplier) for the resources as well. In this market, resource $i$ is priced at $p_i$, $p_i \geq 0$. The firm, starting with a position of $b_i$ units of the $i$ th resource can both buy or sell the resource at the price $p_i$. In this case, the firm is no longer constrained by the resources it has and seeks to solve the following problem instead:

$$\max_{\mathbf{x}\geq\mathbf{0}} \quad \mathbf{c}'\mathbf{x} + \sum_{i=1}^{m} p_i (b_i - \sum_{j=1}^{n} a_{ij}x_j). \tag{6.2}$$

Note that if $b_i > \sum_{j=1}^{n} a_{ij}x_j$, the firm sells $b_i - \sum_{j=1}^{n} a_{ij}x_j$ units of resource $i$ to the market. If $b_i < \sum_{j=1}^{n} a_{ij}x_j$, then the firm buys $\sum_{j=1}^{n} a_{ij}x_j - b_i$ units of resource $i$ from the market.

Given the production decision $\mathbf{x}$ and the market prices $p_i$, we can rewrite the objective in (6.2) as

$$\begin{aligned} & \mathbf{c}'\mathbf{x} + \sum_{i=1}^{m} p_i b_i - \sum_{i=1}^{m} p_i (\sum_{j=1}^{n} a_{ij}x_j) \\ =& \mathbf{c}'\mathbf{x} + \mathbf{p}'\mathbf{b} - \mathbf{p}'\mathbf{A}\mathbf{x} \\ =& \mathbf{p}'\mathbf{b} + (\mathbf{c}' - \mathbf{p}'\mathbf{A})\mathbf{x}. \end{aligned}$$

From the market's perspective:

- The term $\mathbf{p}'\mathbf{b}$ can be interpreted as the buy-back cost from the firm.
- The term $(\mathbf{c}' - \mathbf{p}'\mathbf{A})\mathbf{x} = \sum_{j=1}^{n} (c_j - \mathbf{p}'\mathbf{A}_j)x_j$ can be interpreted as the opportunity cost incurred by the market for not producing the products by itself.

The market then seeks to solve

$$\min_{\mathbf{p}} \mathbf{p}'\mathbf{b} + \left\{ \max_{\mathbf{x}\geq\mathbf{0}} (\mathbf{c}' - \mathbf{p}'\mathbf{A})\mathbf{x} \right\}.$$

- If $c_j - \mathbf{p}'\mathbf{A}_j > 0$ for some $j = 1, .., n$, there exists an arbitrage opportunity for producing product $j$ and $x_j \to \infty$.
- If $c_j - \mathbf{p}'\mathbf{A}_j \leq 0$, then there is no arbitrage opportunity and the profit from producing $j$ is 0.

The market's problem then becomes

$$\begin{aligned} \min \quad & \mathbf{p}'\mathbf{b} \\ s.t. \quad & \mathbf{p}'\mathbf{A} \geq \mathbf{c}' \\ & \mathbf{p}' \geq \mathbf{0}', \end{aligned}$$

which is the dual of (6.1).

## 6.2 Profit Allocation Problem

Consider now there are a set of firms $\mathcal{N} = \{1, 2, ..., N\}$. Firm $i$ owns a resource vector $\mathbf{b}^i$, and a subset of firms $\mathcal{S} \subseteq \mathcal{N}$ can pool their resources to engage in a joint production

$$\begin{aligned} V(\mathcal{S}) = \max \quad & \mathbf{c}'\mathbf{x} \\ s.t. \quad & \mathbf{A}\mathbf{x} \leq \sum_{i \in \mathcal{S}} \mathbf{b}^i \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

The function $V : 2^{\mathcal{N}} \to \mathbb{R}$ is called the *characteristic function*, and we assume here that it takes finite values for any $\mathcal{S} \subseteq \mathcal{N}$. The pair $(\mathcal{N}, V)$ defines a *cooperative game*. In this game, we are concerned with how to allocate the total profit $V(\mathcal{N})$ to each of the firm in $\mathcal{N}$.

An allocation is a vector $\mathbf{l} = (l_1, ..., l_N) \in \mathbb{R}^N$, and is said to be in the *core* if

$$\sum_{i \in \mathcal{S}} l_i \geq V(\mathcal{S})$$

for any $\mathcal{S} \subseteq \mathcal{N}$ and $\sum_{i \in \mathcal{N}} l_i = V(\mathcal{N})$. The problem of finding a core allocation can be formulated as the following linear programming problem:

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{N}} l_i \\ s.t. \quad & \sum_{i \in \mathcal{S}} l_i \geq V(\mathcal{S}), \ \mathcal{S} \subseteq \mathcal{N}. \end{aligned}$$

The above problem has exponential number of constraints and is difficult to solve in general[1]. We have the following result regarding to the core of the game $(\mathcal{N}, V)$.

**Theorem 6.1 (Owen 1975)** *Let $\mathbf{p}^*$ be the optimal solution to the dual problem:*

$$\begin{aligned} \min \quad & \mathbf{p}' \left( \sum_{i \in \mathcal{N}} \mathbf{b}^i \right) \\ s.t. \quad & \mathbf{p}'\mathbf{A} \geq \mathbf{c}' \\ & \mathbf{p}' \geq \mathbf{0}', \end{aligned}$$

*and let $l_i = (\mathbf{p}^*)'\mathbf{b}^i$ for $i = 1, ..., N$. Then, $\mathbf{l}$ is in the core of the game $(\mathcal{N}, V)$.*

[1] It can, however, be solved in polynomial time when $V(\mathcal{S})$ is supermodular.

*Proof:* First note that by strong duality $\sum_{i\in\mathcal{N}} l_i = \sum_{i\in\mathcal{N}}(\mathbf{p}^*)'\mathbf{b}^i = V(\mathcal{N})$. In addition, for any $\mathcal{S} \subseteq \mathcal{N}$, the dual problem to the joint production problem is

$$\begin{aligned} V(\mathcal{S}) = \min \quad & \mathbf{p}'\left(\sum_{i\in\mathcal{S}} \mathbf{b}^i\right) \\ s.t. \quad & \mathbf{p}'\mathbf{A} \geq \mathbf{c}' \\ & \mathbf{p}' \geq \mathbf{0}'. \end{aligned}$$

Clearly, $\mathbf{p}^*$ is a feasible solution to the above problem and hence we must have

$$\sum_{i\in\mathcal{S}} l_i = (\mathbf{p}^*)'\left(\sum_{i\in\mathcal{S}} \mathbf{b}^i\right) \geq V(\mathcal{S}).$$

■

## 6.3 Robust Optimization Problem

Consider the problem (6.1) again—which we refer to as the *nominal problem* here. For any resource constraint $i$:

$$\mathbf{a}_i'\mathbf{x} \leq b_i,$$

instead of knowing the resource requirement $a_{ij}$ for each of the product $j$ exactly, we assume now that it is uncertain and we use $\tilde{a}_{ij}$ to denote this random variable. Suppose $\tilde{a}_{ij}$ has support $[a_{ij} - \delta_{ij}, a_{ij} + \delta_{ij}]$, where $\delta_{ij} \geq 0$. We can hence alternatively write

$$\tilde{a}_{ij} = a_{ij} + \delta_{ij} z_{ij},$$

where $z_{ij}$ is a random variable with support $[-1, 1]$.

### Model of Soyster (1973)

In this model, we require our decisions to satisfy the constraint

$$\sum_{j=1}^{n} \tilde{a}_{ij} x_j \leq b_i,$$

for any possible realizations of $\tilde{a}_{ij}$. Equivalently, we require

$$g_i(\mathbf{x}) \leq b_i,$$

where

$$\begin{aligned} g_i(\mathbf{x}) = \max_{z_{ij}} \quad & \sum_{j=1}^{n} (a_{ij} + \delta_{ij} z_{ij}) x_j \\ s.t. \quad & -1 \leq z_{ij} \leq 1. \end{aligned}$$

Note that $\max_{-1\leq z_{ij}\leq 1} \delta_{ij} x_j z_{ij} = \delta_{ij}|x_j|$. Hence, $g_i(\mathbf{x}) = \sum_{j=1}^{n}(a_{ij}x_j + \delta_{ij}|x_j|)$, and we have a *robust counterpart* of problem (6.1)

$$\begin{aligned} \max \quad & \mathbf{c}'\mathbf{x} \\ s.t. \quad & \sum_{j=1}^{n} (a_{ij} x_j + \delta_{ij}|x_j|) \leq b_i, \; i = 1, ..., m \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

which can be reformulated as the following LP[2]:

$$
\begin{aligned}
\max \quad & \mathbf{c}'\mathbf{x} \\
s.t. \quad & \sum_{j=1}^{n}(a_{ij}x_j + \delta_{ij}y_j) \leq b_i,\ i = 1, ..., m \\
& x_j \leq y_j, \\
& -x_j \leq y_j, \\
& \mathbf{x} \geq \mathbf{0}.
\end{aligned}
$$

## Model of Bertsimas and Sim (2004)

Note that in solving for $g_i(\mathbf{x})$ in the model of Soyster (1973), we must have at optimum $\sum_{j=1}^{n} |z_{ij}| = n$. In other words, either $z_{ij} = 1$ or $z_{ij} = -1$ and all random variables achieve the worst case simultaneously—which is highly unlikely in practice.

To model certain risk pooling effect, we can require

$$\sum_{j=1}^{n} |z_{ij}| \leq \alpha_i n$$

for some $\alpha_i \in [0, 1]$ and we find the worst case realization by solving

$$
\begin{aligned}
g_i(\mathbf{x}) = \max_{z_{ij}} \quad & \sum_{j=1}^{n}(a_{ij} + \delta_{ij}z_{ij})x_j \\
s.t. \quad & -1 \leq z_{ij} \leq 1 \\
& \sum_{j=1}^{n} |z_{ij}| \leq \alpha_i n.
\end{aligned}
$$

- When $\alpha_i = 0$, $g_i(\mathbf{x}) = \mathbf{a}_i'\mathbf{x}$ and we are back to the nominal problem.
- When $\alpha_i = 1$, the constraint $\sum_{j=1}^{n} |z_{ij}| \leq \alpha_i n$ is redundant and we are back to the model of Soyster (1973).

Note that at optimum, $z_{ij}$ must have the same sign as $x_j$ (otherwise be flipping the sign of $z_{ij}$, we still have a feasible solution with higher objective value), i.e., $z_{ij}x_j = |z_{ij}||x_j|$. It follows that $g_i(\mathbf{x})$ can be reformulated as the linear program

$$
\begin{aligned}
g_i(\mathbf{x}) = \sum_{j=1}^{n} a_{ij}x_j + \max_{z_{ij}} \quad & \sum_{j=1}^{n} \delta_{ij}|x_j|z_{ij} \\
s.t. \quad & z_{ij} \leq 1 \\
& \sum_{j=1}^{n} z_{ij} \leq \alpha_i n \\
& z_{ij} \geq 0.
\end{aligned}
\tag{6.3}
$$

[2]Of course, since we know $\mathbf{x} \geq \mathbf{0}$, the term $|x_j|$ can be directly replaced by $x_j$ without using the reformulation. But the reformulation works for more general problems.

In general, we do not have a closed-form for $g_i(\mathbf{x})$ (what kind of function it is?). The question now is how we can solve the robust counterpart of problem (6.1):

$$\begin{aligned} \max \quad & \mathbf{c}'\mathbf{x} \\ s.t. \quad & g_i(\mathbf{x}) \leq b_i, \ i = 1, ..., m \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

without an explicit expression for $g_i(\mathbf{x})$. Note that $g_i(\mathbf{x}) \leq b_i$ if and only if for the optimal solution to (6.3): $z_{ij}^*$, i.e., the worst case realization, we have $\sum_{j=1}^n a_{ij}x_j + \sum_{j=1}^n \delta_{ij}|x_j|z_{ij}^* \leq b_i$. The difficulty here is $z_{ij}^*$ depends on $\mathbf{x}$.

Consider the dual problem of (6.3):

$$\begin{aligned} g_i(\mathbf{x}) = \sum_{j=1}^n a_{ij}x_j + \min_{p_{ij},\lambda_i} \quad & \sum_{j=1}^n p_{ij} + \alpha_i n \lambda_i \\ s.t. \quad & p_{ij} + \lambda_i \geq \delta_{ij}|x_j|, \ j = 1, ..., n \\ & p_{ij}, \lambda_i \geq 0. \end{aligned}$$

The key observation here is that $g_i(\mathbf{x}) \leq b_i$ if and only if there exists $p_{ij}, \lambda_i \geq 0$ and $p_{ij} + \lambda_i \geq \delta_{ij}|x_j|$ for $j = 1, ..., n$ such that

$$\sum_{j=1}^n a_{ij}x_j + \sum_{j=1}^n p_{ij} + \alpha_i n \lambda_i \leq b_i.$$

As a result, the robust counterpart can be reformulated as

$$\begin{aligned} \max \quad & \mathbf{c}'\mathbf{x} \\ s.t. \quad & \sum_{j=1}^n a_{ij}x_j + \sum_{j=1}^n p_{ij} + \alpha_i n \lambda_i \leq b_i, \ i = 1, ..., m \\ & p_{ij} + \lambda_i \geq \delta_{ij}|x_j|, \ i = 1, ..., m, j = 1, ..., n \\ & \mathbf{x} \geq \mathbf{0}, p_{ij}, \lambda_i \geq 0, \end{aligned}$$

which can be reformulated as an LP:

$$\begin{aligned} \max \quad & \mathbf{c}'\mathbf{x} \\ s.t. \quad & \sum_{j=1}^n a_{ij}x_j + \sum_{j=1}^n p_{ij} + \alpha_i n \lambda_i \leq b_i, \ i = 1, ..., m \\ & p_{ij} + \lambda_i \geq \delta_{ij}y_j, \ i = 1, ..., m, j = 1, ..., n \\ & y_j \geq x_j, \ y_j \geq -x_j, \ j = 1, ..., n \\ & \mathbf{x} \geq \mathbf{0}, p_{ij}, \lambda_i \geq 0. \end{aligned}$$

## References

[Bertsimas and Sim 2004] D. Bertsimas and M. Sim, 2004, "The price of robustness", *Operations Research*, 52(1), pp. 35–53.

[Owen 1975] G. Owen, 1975, "On the core of linear production games", *Mathematical Programming*, 9(1), pp. 358–370.

[Soyster 1973] A.L Soyster, 1973, "Convex programming with set-inclusive constraints and applications to inexact linear programming", *Operations Research*, 21, pp. 1154–1157.

**BDC6111: Introduction to Optimization** **Fall 2018, NUS**

# Lecture 7: Large Scale Optimization

*Lecturer: Zhenyu Hu*

## 7.1 Revised Simplex Method and Delayed Column Generation

Consider the standard form problem

$$\begin{aligned} \min \quad & \mathbf{c}'\mathbf{x} \\ s.t. \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

In the full tableau implementation, in each iteration we update the following table which requires us to

| $-\mathbf{c}_B'\mathbf{B}^{-1}\mathbf{b}$ | $\mathbf{c}' - \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A}$ |
|---|---|
| $\mathbf{B}^{-1}\mathbf{b}$ | $\mathbf{B}^{-1}\mathbf{A}$ |

compute the reduced cost $\bar{c}_j = c_j - \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A}_j$ and the search direction $\mathbf{B}^{-1}\mathbf{A}_j$ for all nonbasic variables $x_j$.

However, in some large scale optimization problems, we have a huge number of decision variables, i.e, $n$ is very large, and accessing every column in the matrix $\mathbf{A}$ in each iteration can be time consuming. One can overcome this difficulty if the following two steps can be achieved:

- One can efficiently solve
$$\min_{j=1,\ldots,n} \bar{c}_j$$
without computing every $\bar{c}_j$;
- One can perform simplex iteration without accessing to each column $\mathbf{A}_i$ in each iteration.

The second step is the key idea behind the *revised simplex method*, whose typical iteration is summarized below.

**An iteration of the revised simplex method**

1. In each iteration, we start with the basic columns $\mathbf{A}_{B(1)}, \ldots\mathbf{A}_{B(m)}$, the associated BFS $\mathbf{x}$ and $\mathbf{B}^{-1}$.
2. Compute the reduced costs $\bar{c}_j = c_j - \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A}_j$ sequentially. If one encounters $\bar{c}_j < 0$ for some $j$ for the first time, then stop and return the index $j$. If all reduced costs are all nonnegative, the current basic feasible solution is optimal, and the algorithm terminates.
3. For the returned nonbasic index $j$, compute $\mathbf{u} = \mathbf{B}^{-1}\mathbf{A}_j$. If $\mathbf{u} \leq \mathbf{0}$, then the optimal cost is $-\infty$ and the algorithm terminates; else let
$$\theta^* = \min_{\{i|u_i>0\}} \frac{x_{B(i)}}{u_i}.$$
and $l$ be the index such that $\theta^* = \frac{x_{B(l)}}{u_l}$. Form a new basis by replacing $\mathbf{A}_{B(l)}$ with $\mathbf{A}_j$ and compute the new BFS $\mathbf{y}$ via $y_j = \theta^*$, and $y_{B(i)} = x_{B(i)} - \theta^* u_i$ for $i \neq l$.

4. Finally, we update $\mathbf{B}^{-1}$ to $\bar{\mathbf{B}}^{-1}$ by performing row operations to make
$$\begin{bmatrix} \mathbf{B}^{-1} & \mathbf{u} \end{bmatrix} \rightarrow \begin{bmatrix} \bar{\mathbf{B}}^{-1} & \mathbf{e}_l \end{bmatrix}.$$

Note that the steps 3 and 4 are essentially the same as solving

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} c_{B(i)} x_{B(i)} + c_j x_j \\ s.t. \quad & \mathbf{B}\mathbf{x}_B + \mathbf{A}_j x_j = \mathbf{b}, \\ & \mathbf{x}_B \geq \mathbf{0}, x_j \geq 0. \end{aligned}$$

In some delayed column generation methods, instead of keeping just the basic columns and throwing away the exit column in each iteration, one may keep some of the columns $\{\mathbf{A}_i | i \in I\}$ with $I \subseteq \{1, ..., n\}$ and solve the following smaller problems (without explicitly going through the simplex iteration as in steps 3 and 4)

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} c_{B(i)} x_{B(i)} + c_j x_j + \sum_{i \in I} c_i x_i \\ s.t. \quad & \mathbf{B}\mathbf{x}_B + \mathbf{A}_j x_j + \sum_{i \in I} \mathbf{A}_i x_i = \mathbf{b}, \\ & \mathbf{x}_B \geq \mathbf{0}, x_i, x_j \geq 0. \end{aligned}$$

In the revised simplex method, once a column $\mathbf{A}_j$ with negative reduced cost is found, the rest of the nonbasic columns will not be accessed when performing steps 3 and 4. However, in step 2, in the worst case, one still needs to generate every column $\mathbf{A}_j$—the generation of the columns is not delayed.

We demonstrate using the example below that when the problem has certain special structure, $\min_j \bar{c}_j$ can be computed without accessing to every column $\mathbf{A}_j$.

**Example 7.1 (Cutting Stock Problem)** *Consider a paper company that has a supply of large rolls of paper of width $W$, which is assumed to be a positive integer. There are demands for $b_i$ rolls of paper with width $w_i$, where $w_i \leq W$ for $i = 1, ..., m$. A large roll can be sliced in a certain pattern to obtain smaller rolls. Let $a_i$ be the number of rolls of width $w_i$ to be produced from a single large roll. A feasible pattern $(a_1, ..., a_m)$ then must satisfy*

$$\sum_{i=1}^{m} a_i w_i \leq W.$$

*If there are in total $n$ feasible patterns, we then collect all feasible patterns in a matrix $\mathbf{A}$ of dimension $m \times n$.*

*For instance, when $W = 7, w_1 = 2, w_2 = 4$, the following matrix summarizes all feasible patterns:*

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 1 & 2 & 3 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix},$$

*with the column $\mathbf{A}_j$ corresponding to a pattern $j$.*

*Let $x_j$ be the number of large rolls cut according to pattern $j$. The company seeks to minimize the number of large rolls used while satisfying customer demand:*

$$\begin{aligned} \min \quad & \sum_{j=1}^{n} x_j \\ s.t. \quad & \sum_{j=1}^{n} a_{ij} x_j = b_i, i = 1, ..., m \\ & x_j \geq 0, j = 1, ..., n. \end{aligned}$$

*Each $x_j$ should be an integer, but we consider its linear relaxation here.*

*Even if $m$ is small, $n$ can be huge and even writing down all the columns of $\mathbf{A}$ can be difficult. Instead, we can use revised simplex method and start with the basis $\mathbf{I}$, i.e., the pattern where only one roll of width $w_j$ is produced from a large roll is always feasible (although may not be economical) for any $j = 1, ..., m$.*

*At each iteration, given a basis $\mathbf{B}$, we can compute the dual basic solution $\mathbf{p}' = \mathbf{c}_B'\mathbf{B}^{-1}$. Instead of computing $\bar{c}_j = c_j - \mathbf{p}'\mathbf{A}_j = 1 - \mathbf{p}'\mathbf{A}_j$ for every $j = 1, ..., n$. We seek to solve $\min_j \bar{c}_j$ or equivalently $\max_j \mathbf{p}'\mathbf{A}_j$. By definition of $\mathbf{A}_j, j = 1, ..., n$, this is equivalent as*

$$\begin{aligned} \max_{a_i, i=1,\ldots,m} \quad & \sum_{i=1}^{m} p_i a_i \\ s.t. \quad & \sum_{i=1}^{n} w_i a_i \le W, \\ & a_i \ge 0, \text{ integer}, \ i = 1, ..., m. \end{aligned}$$

*The above problem is called the* knapsack problem*—although known as an NP-hard problem, can still be efficiently solved when $m$ is small.*

*Upon solving the knapsack problem:*

- *If the knapsack problem returns an optimal value less than or equal to one, we then know $\bar{c}_j \ge \min_i \bar{c}_i \ge 0$. Hence, the current basis $\mathbf{B}$ is optimal.*
- *If the knapsack problem returns a value greater than one with optimal solution $(a_1^*, ..., a_m^*)$, then we have identified a nonbasic column $\mathbf{A}_j' = (a_1^*, ..., a_m^*)$, that enters the basis.*

## 7.2 Delayed Constraint Generation

Consider the dual of the standard form problem

$$\begin{aligned} \max \quad & \mathbf{p}'\mathbf{b} \\ s.t. \quad & \mathbf{p}'\mathbf{A}_i \le c_i, i = 1, ..., n. \end{aligned}$$

When $\mathbf{A}$ has large number of columns, i.e., $n$ is large, the number of constraints in the above dual problem is large. Like delayed column generation, we can consider a subset $I \subseteq \{1, ..., n\}$ of the constraints, and form the relaxed dual problem

$$\begin{aligned} \max \quad & \mathbf{p}'\mathbf{b} \\ s.t. \quad & \mathbf{p}'\mathbf{A}_i \le c_i, i \in I. \end{aligned}$$

Let $\mathbf{p}^*$ be the optimal basic feasible solution to the relaxed dual problem.

- If $\mathbf{p}^*$ satisfies all the constraints $\mathbf{p}'\mathbf{A}_i \le c_i, i = 1, ..., n$. Then, $\mathbf{p}^*$ must also be optimal to the original dual problem, and the algorithm terminates.
- If $\mathbf{p}^*$ violates constraint $i$ for some $i \notin I$, then we add $i$ into $I$.

The step of checking feasibility is the same as checking the nonnegativity of the reduced cost in the delayed column generation method, and we need an efficient method for identifying a violated constraint. Usually, this is achieved by finding an efficient way for solving

$$\min_{i=1,\ldots,n} c_i - (\mathbf{p}^*)'\mathbf{A}_i.$$

Solving the above problem without going through every term $c_i - (\mathbf{p}^*)'\mathbf{A}_i$ is possible when the problem has certain special structure, which we demonstrate next.

## 7.3 Stochastic Programming and Benders Decomposition

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Consider a decision maker who acts in two consecutive stages with some random information being revealed in the second stage. In the first stage, the decision maker needs to choose a vector $\mathbf{x}$ that satisfies the constraints

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

The decision $\mathbf{x}$ generates an immediate cost $\mathbf{c}'\mathbf{x}$.

In the second stage, some random variables $\mathbf{B}(\omega), \mathbf{d}(\omega)$ are revealed, where $\omega$ denotes a particular scenario (or sample) from the sample space $\Omega$. Given a particular scenario $\omega$ and the first stage decision $\mathbf{x}$, the decision maker needs to choose another vector $\mathbf{y}(\omega)$ that satisfies the constraints

$$\begin{aligned} &\mathbf{B}(\omega)\mathbf{x} + \mathbf{D}\mathbf{y}(\omega) = \mathbf{d}(\omega), \\ &\mathbf{y}(\omega) \geq \mathbf{0}. \end{aligned}$$

The decision $\mathbf{y}(\omega)$ generates a second stage cost $\mathbf{f}'\mathbf{y}(\omega)$. Let $z(\mathbf{x}, \omega)$ be the minimum second stage cost given a scenario $\omega$ and first stage decision $\mathbf{x}$. It follows that

$$\begin{aligned} z(\mathbf{x}, \omega) = \min_{\mathbf{y}(\omega)} \; & \mathbf{f}'\mathbf{y}(\omega) \\ \text{s.t. } & \mathbf{B}(\omega)\mathbf{x} + \mathbf{D}\mathbf{y}(\omega) = \mathbf{d}(\omega), \\ & \mathbf{y}(\omega) \geq \mathbf{0}. \end{aligned} \tag{7.1}$$

Now, the optimization problem in the first stage can be written as

$$\begin{aligned} \min_{\mathbf{x}} \;\; & \mathbf{c}'\mathbf{x} + \mathbb{E}_{\mathbb{P}}[z(\mathbf{x}, \omega)] \\ \text{s.t. } & \mathbf{Ax} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned} \tag{7.2}$$

While $\mathbb{E}_{\mathbb{P}}[z(\mathbf{x}, \omega)]$ is in general a nonlinear function of $\mathbf{x}$, the above problem can nevertheless be formulated as an LP when $\Omega$ consists of finite samples, say, $\omega_1, ..., \omega_K$. Let $\alpha_i$ be the probability of scenario $\omega_i$. The above problem is then equivalent to

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}_i, i=1,...,K} \;\; & \mathbf{c}'\mathbf{x} + \sum_{i=1}^{K} \alpha_i \mathbf{f}'\mathbf{y}_i \\ \text{s.t. } & \mathbf{Ax} = \mathbf{b}, \\ & \mathbf{B}_i\mathbf{x} + \mathbf{D}\mathbf{y}_i = \mathbf{d}_i, \;\; i = 1, ..., K \\ & \mathbf{x}, \mathbf{y}_1, ..., \mathbf{y}_K \geq \mathbf{0}. \end{aligned} \tag{7.3}$$

**Example 7.2 (Joint Inventory and Transportation Problem)** *Suppose a retailer manages the inventory at $n$ warehouses, which are used to satisfy random demands at $m$ locations. In the first stage, the retailer needs to decide $\mathbf{x} \in \mathbb{R}^n$ with $x_i$ being the inventory placed at warehouse $i$ for $i = 1, ..., n$. The procurement cost at warehouse $i$ is $c_i$ so that the total procurement cost generated in the first stage is $\mathbf{c}'\mathbf{x}$.*

*In the second stage, the demand $\mathbf{d}(\omega)$ at $m$ locations is realized with $d_j(\omega)$ being the demand at location $j$ in scenario $\omega$. Given the inventory level $\mathbf{x}$ and the demand realization $\mathbf{d}(\omega)$, the retailer needs to decide $y_{ij}(\omega)$, the amount of inventory transported from warehouse $i$ to satisfy demand at location $j$. The unit transportation cost from $i$ to $j$ is $t_{ij}$ and the unit revenue for satisfying demand at location $j$ is $r_j$. The second stage problem is then*

$$\begin{aligned} z(\mathbf{x}, \omega) = &\min_{\mathbf{y}(\omega)} \sum_{i=1}^{n} \sum_{j=1}^{m} (t_{ij} - r_j) y_{ij}(\omega) \\ &s.t. \ \sum_{j=1}^{m} y_{ij}(\omega) \le x_i, \\ &\quad \sum_{i=1}^{n} y_{ij}(\omega) \le d_j(\omega), \\ &\quad \mathbf{y}(\omega) \ge \mathbf{0}. \end{aligned}$$

*The first stage problem is simply*

$$\begin{aligned} &\min_{\mathbf{x}} \ \mathbf{c}'\mathbf{x} + \mathbb{E}_{\mathbb{P}}[z(\mathbf{x}, \omega)] \\ &s.t. \ \mathbf{x} \ge \mathbf{0}. \end{aligned}$$

When $K$ is large and $\mathbf{y}(\omega)$, $\mathbf{d}(\omega)$ has dimension $m, t$ respectively, the formulation in (7.3) is an LP with $O(mK)$ decision variables and $O(tK)$ equality constraints, and can be computationally demanding to solve[1]. Observe that given a fixed $\mathbf{x}$, the problems of finding $\mathbf{y}(\omega)$ are all decoupled and we can solve $K$ much smaller LPs, i.e., (7.1), with $m$ decision variables and $t$ equality constraints. The difficulty lies in the fact that finding $\mathbf{x}$ is coupled with finding $\mathbf{y}(\omega), \omega \in \Omega$. The idea behind Benders decomposition is to decouple the two tasks.

In the following we assume that (7.1) is feasible and has finite optimal value for any $\omega \in \Omega$. The dual of (7.1) is

$$\begin{aligned} z(\mathbf{x}, \omega) = &\max_{\mathbf{p}(\omega)} \mathbf{p}'(\omega)(\mathbf{d}(\omega) - \mathbf{B}(\omega)\mathbf{x}) \\ &\text{s.t. } \mathbf{p}'(\omega)\mathbf{D} \le \mathbf{f}'. \end{aligned} \tag{7.4}$$

Let $\mathbf{p}^i, i = 1, ..., I$ be the extreme points of $\{\mathbf{p} | \mathbf{p}'\mathbf{D} \le \mathbf{f}'\}$. By our assumption on (7.1), problem (7.4) also has finite optimal value and we must have

$$z(\mathbf{x}, \omega) = \max_{i=1,\ldots,I} (\mathbf{p}^i)'(\mathbf{d}(\omega) - \mathbf{B}(\omega)\mathbf{x}),$$

which is equivalent to

$$\begin{aligned} z(\mathbf{x}, \omega) = &\min_{z(\omega)} z(\omega) \\ &\text{s.t. } (\mathbf{p}^i)'(\mathbf{d}(\omega) - \mathbf{B}(\omega)\mathbf{x}) \le z(\omega), \ i = 1, ..., I. \end{aligned}$$

We can then reformulate (7.3) as

$$\begin{aligned} &\min_{\mathbf{x}, z(\omega)} \ \mathbf{c}'\mathbf{x} + \sum_{\omega \in \Omega} \alpha_i z(\omega) \\ &\text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}, \\ &\quad (\mathbf{p}^i)'(\mathbf{d}(\omega) - \mathbf{B}(\omega)\mathbf{x}) \le z(\omega), \ i = 1, ..., I, \omega \in \Omega, \\ &\quad \mathbf{x} \ge \mathbf{0}. \end{aligned} \tag{7.5}$$

[1]Keep in mind that computing the inverse of an $m \times m$ dimension matrix $\mathbf{B}$ or solving a linear system $\mathbf{B}\mathbf{x} = \mathbf{b}$ takes $O(m^3)$.

We call formulation (7.5) the *master problem*, which only has $O(K)$ decision variables (as opposed to $O(mK)$ in (7.3)). But (7.5) has an extremely large number of inequality constraints—$O(IK)$. We can overcome this via delayed constraint generation.

We start with (7.3) that involves only a subset of inequality constraints. Suppose the resulting optimal solution to this relaxed master problem is $\mathbf{x}^*$ and $\mathbf{z}^* = (z_1^*, ..., z_K^*)$. We then need to check the feasibility of $(\mathbf{x}^*, \mathbf{z}^*)$ with respect to the rest of constraints in (7.3). The key idea here is to solve some auxiliary subproblems instead of checking the constraints $(\mathbf{p}^i)'(\mathbf{d}(\omega) - \mathbf{B}(\omega)\mathbf{x}^*) \leq z^*(\omega)$ one by one. In particular, for each $\omega \in \Omega$, we solve

$$\begin{aligned} &\min_{\mathbf{y}(\omega)} \mathbf{f}'\mathbf{y}(\omega) \\ \text{s.t. } &\mathbf{D}\mathbf{y}(\omega) = \mathbf{d}(\omega) - \mathbf{B}(\omega)\mathbf{x}^*, \\ &\mathbf{y}(\omega) \geq \mathbf{0}. \end{aligned}$$

From solving above problem, we can obtain the optimal dual BFS: $\mathbf{p}^{i(\omega)}$ for every $\omega \in \Omega$.

- If $(\mathbf{p}^{i(\omega)})'(\mathbf{d}(\omega) - \mathbf{B}(\omega)\mathbf{x}^*) \leq z^*(\omega)$ for every $\omega \in \Omega$, then by optimality of $\mathbf{p}^{i(\omega)}$,

$$(\mathbf{p}^i)'(\mathbf{d}(\omega) - \mathbf{B}(\omega)\mathbf{x}^*) \leq z^*(\omega)$$

  for all $i = 1, ..., I$. As a result, $(\mathbf{x}^*, \mathbf{z}^*)$ is feasible to (7.5) and hence optimal.

- If $(\mathbf{p}^{i(\bar{\omega})})'(\mathbf{d}(\bar{\omega}) - \mathbf{B}(\bar{\omega})\mathbf{x}^*) > z^*(\bar{\omega})$ for some $\bar{\omega} \in \Omega$, then we have identified a violating constraint:

$$(\mathbf{p}^{i(\bar{\omega})})'(\mathbf{d}(\bar{\omega}) - \mathbf{B}(\bar{\omega})\mathbf{x}) \leq z(\bar{\omega}),$$

  which is added to the relaxed master problem.

# References

[BT97] D. Bertsimas and J.N. Tsitsiklis, *Introduction to Linear Optimization*, Springer, 1997.

**BDC6111: Introduction to Optimization** **Fall 2018, NUS**

# Lecture 8: Network Flow Problems

*Lecturer: Zhenyu Hu*

## 8.1 Min-Cost Network Flow Problem

Given a directed connected graph $G = (\mathcal{N}, \mathcal{A})$, we associate with each node $i \in \mathcal{N}$ an integer number $b_i$.

- If $b_i > 0$, the node $i$ is called a *supply* or *source* node;
- If $b_i < 0$, the node $i$ is called a *demand* or *sink* node;
- If $b_i = 0$, the node $i$ is called a *transshipment* node.

We assume that

$$\sum_{i \in \mathcal{N}} b_i = 0.$$

The general network flow problem concerns with sending materials from the source node to the sink node through the arcs of the network at minimum cost. Along the arc $(i, j)$, the cost per unit of flow is $c_{ij}$ and the maximum units of flow is $u_{ij}$. The min-cost network flow problem can be formulated as the following linear programming problem:

$$\begin{aligned} \min \quad & \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij} \\ s.t. \quad & \sum_{j:(i,j) \in \mathcal{A}} x_{ij} - \sum_{j:(j,i) \in \mathcal{A}} x_{ji} = b_i, \; i \in \mathcal{N} \\ & 0 \leq x_{ij} \leq u_{ij}, (i,j) \in \mathcal{A}. \end{aligned}$$

### Transportation Problem

In this case $G$ is a complete bipartite graph with $m$ supply nodes and $n$ demand nodes. Let

- $a_i$ be number of units available at supply $i$, $i = 1, ..., m$;
- $b_j$ be the number of units required at demand $j$, $j = 1, ..., n$;
- $c_{ij}$ be the unit transportation cost from $i$ to $j$.

We have

$$\begin{aligned}
\min \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \\
s.t. \quad & \sum_{j=1}^{n} x_{ij} = a_i, \ i = 1, ..., m \\
& \sum_{i=1}^{n} x_{ij} = b_j, \ j = 1, ..., n \\
& x_{ij} \geq 0.
\end{aligned}$$

## Assignment Problem

In the transportation problem when $m = n$ and $a_i = b_j = 1$, we then have the assignment problem:

$$\begin{aligned}
\min \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\
s.t. \quad & \sum_{j=1}^{n} x_{ij} = 1, \ i = 1, ..., n \\
& \sum_{i=1}^{n} x_{ij} = 1, \ j = 1, ..., n \\
& x_{ij} \geq 0.
\end{aligned}$$

## Shortest Path Problem

The problem concerns with finding the minimum cost path from a source node $s$ to a sink node $t$, with $c_{ij}$ being the cost associated with traversing the arc $(i, j)$. The problem can be formulated as

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} \\
s.t. \quad & \sum_{j:(i,j)\in\mathcal{A}} x_{ij} - \sum_{j:(j,i)\in\mathcal{A}} x_{ji} = \begin{cases} 1, & \text{if } i = s, \\ -1, & \text{if } i = t, \\ 0, & \text{otherwise,} \end{cases} \\
& x_{ij} \geq 0, (i, j) \in \mathcal{A}.
\end{aligned}$$

## Maximum Flow Problem

The problem is to find the largest possible amount of flow that can be sent through the network, from $s$ to $t$ with the capacity on the arc $(i,j)$ being $u_{ij}, u_{ij} \geq 0$. The problem can be formulated as

$$\begin{aligned} \max \quad & v \\ s.t. \quad & \sum_{j:(i,j)\in\mathcal{A}} x_{ij} - \sum_{j:(j,i)\in\mathcal{A}} x_{ji} = \begin{cases} v, & \text{if } i = s, \\ -v, & \text{if } i = t, \\ 0, & \text{otherwise,} \end{cases} \\ & 0 \leq x_{ij} \leq u_{ij}, (i,j) \in \mathcal{A}. \end{aligned} \tag{8.1}$$

For any feasible flow, the objective value $v$ is called the *value* of the flow. Note that $v$ is a decision variable itself, and hence the above formulation is not an exact match to the min-cost network flow problem. By adding an artificial arc $(t,s)$ with $u_{ts} = +\infty$ to $\mathcal{A}$, i.e., $\tilde{\mathcal{A}} = \mathcal{A} \cup \{(t,s)\}$, we can reformulate the above problem as

$$\begin{aligned} \max \quad & x_{ts} \\ s.t. \quad & \sum_{j:(i,j)\in\tilde{\mathcal{A}}} x_{ij} - \sum_{j:(j,i)\in\tilde{\mathcal{A}}} x_{ji} = 0 \\ & 0 \leq x_{ij} \leq u_{ij}, (i,j) \in \tilde{\mathcal{A}}. \end{aligned}$$

## Special Structure

The flow conservation constraints can be concisely written as

$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

where $\mathbf{A}$ is referred to as the *node-arc incidence matrix.* The matrix $\mathbf{A}$ has dimension $|\mathcal{N}| \times |\mathcal{A}|$ and its $(i,k)$-th entry $a_{ik}$ has the property:

$$a_{ik} = \begin{cases} 1, & \text{if } i \text{ is the start node of the } k\text{-th arc,} \\ -1, & \text{if } i \text{ is the end node of the } k\text{-th arc,} \\ 0, & \text{otherwise.} \end{cases}$$

**Example 8.1** *Consider the directed graph in Figure 8.1 below.*

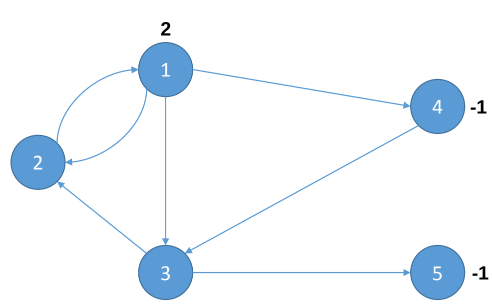



Figure 8.1: A directed graph $(\mathcal{N}, \mathcal{A})$

*Let the flow on the arcs be* $x_{12}, x_{13}, x_{14}, x_{21}, x_{31}, x_{35}, x_{43}$*. The corresponding node-arc incidence matrix is*

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & -1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 1 & -1 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}.$$

Note that each column in matrix $\mathbf{A}$ has exactly two nonzero entries, one equal to $+1$, and one equal to $-1$. In particular, this implies that the sum of all the rows is a zero vector. The special structure here guarantees that the matrix $\mathbf{A}$ is *totally unimodular*—all its square submatrix has determinant $0, \pm 1$ (see Section 4.2 in [CCZ14] for the arguments behind).

Recall for standard form linear programming problem, any basic feasible solution with basis $\mathbf{B}$ is solved via $\mathbf{B}\mathbf{x}_B = \mathbf{b}$. By Carmer's Rule,

$$x_{B(i)} = \frac{\det(\mathbf{M}_i)}{\det(\mathbf{B})},$$

where $\mathbf{M}_i = [\mathbf{B}_1, ..., \mathbf{B}_{i-1}, \mathbf{b}, \mathbf{B}_{i+1}, ..., \mathbf{B}_m]$. As long as, the entries of $\mathbf{b}, \mathbf{A}$ are integers, then $\det(\mathbf{M}_i)$ is an integer for any $i$. If $\mathbf{A}$ is totally unimodular, then $\det(\mathbf{B})$ is either 1 or $-1$ and hence $x_{B(i)}$ must be integer.

Finally, since the summation of the row vectors of $\mathbf{A}$ is a zero vector, the rows are linearly dependent and we must have $\text{rank}(\mathbf{A}) \leq |\mathcal{N}| - 1$.

## 8.2 Spanning Tree and Basic Solution

We consider the uncapacitated min-cost network flow problem here:

$$\begin{aligned} \min \quad & \sum_{(i,j)\in\mathcal{A}} c_{ij} x_{ij} \\ s.t. \quad & \sum_{j:(i,j)\in\mathcal{A}} x_{ij} - \sum_{j:(j,i)\in\mathcal{A}} x_{ji} = b_i, \ i \in \mathcal{N} \\ & x_{ij} \geq 0, (i,j) \in \mathcal{A}, \end{aligned}$$

which is in standard form.

An undirected graph $G = (\mathcal{N}, \mathcal{A})$ is called a *tree* if it is connected and has no cycles. Given a connected undirected graph $G = (\mathcal{N}, \mathcal{A})$, the *spanning tree* of $G$ is a tree $T$ that contains every node in $G$, i.e., $T = (\mathcal{N}, \mathcal{E})$ for some $\mathcal{E} \subseteq \mathcal{A}$.

Given a connected undirected (weighted) graph $G = (\mathcal{N}, \mathcal{A})$, one can generate a (minimum) spanning tree using Prim's algorithm.

1. Initialize a tree with a single vertex.
2. Among the arcs that connect the tree to vertices not yet in the tree, find one (the minimum one for weighted graph) and add it to the tree.
3. Repeat step 2 until all vertices are in the tree.

For example, one spanning tree for the graph in Figure 8.1 (ignoring directions) can be found as in Figure 8.2. We remark the following properties of a spanning tree.

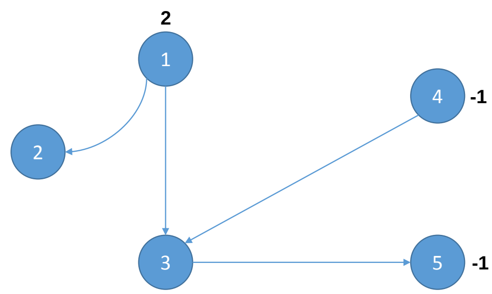



Figure 8.2: Spanning tree for the graph in Figure 8.1

- Given a spanning tree $T = (\mathcal{N}, \mathcal{E})$ of $G = (\mathcal{N}, \mathcal{A})$, one must have $|\mathcal{E}| = |\mathcal{N}| - 1$.
- Conversely, if for an arc set $\mathcal{E} \subseteq \mathcal{A}$ with $|\mathcal{E}| = |\mathcal{N}| - 1$ such that $T = (\mathcal{N}, \mathcal{E})$ is connected, then $T = (\mathcal{N}, \mathcal{E})$ is a spanning tree.

For min-cost network flow problem, one can construct a solution using a spanning tree $T = (\mathcal{N}, \mathcal{E})$ of $G = (\mathcal{N}, \mathcal{A})$ as follows.

1. Let $x_{ij} = 0$ for every $(i, j) \notin \mathcal{E}$.
2. Starting from each leaf of the tree (which has degree one), solve the flow on the arc with its parent node (which is unique) using flow conservation constraint.
3. For any node whose flows on the arcs with all its children are all determined, solve the flow on the arc with its parent node (which is unique) using flow conservation constraint.
4. Repeat step 3 until the root of the tree is reached.

We call the flow $\mathbf{x}$ solved via the above procedure as the *spanning tree solution*. The following theorem demonstrates that spanning tree solution is nothing but the basic solution we defined for general LP problems.

**Theorem 8.2** *A flow vector* $\mathbf{x}$ *is a basic solution if and only if it is a spanning tree solution.*

*Proof:* Suppose $\mathbf{x}$ is a spanning tree solution generated according to the algorithm outlined above. From the way it is constructed, one can see that it satisfies $\mathbf{Ax} = \mathbf{b}$ and it is unique. In other words, $x_{ij} = 0$ for $(i, j) \notin \mathcal{E}$ can be viewed as nonbasic variables and the variables $x_{ij}$ for $(i, j) \in \mathcal{E}$ are the basic variables. The columns corresponding to the basic variables constitute a matrix with $|\mathcal{N}|$ rows and $|\mathcal{N}| - 1$ columns, which has rank $|\mathcal{N}| - 1$, since the system of equations yields a unique solution.

Conversely, if $\mathbf{x}$ is a basic solution, then let $\mathcal{E} = \{(i, j) | x_{ij} \text{ is a basic variable}\}$. By definition, for $(i, j) \notin \mathcal{E}$, $x_{ij} \neq 0$ and $|\mathcal{E}| = |\mathcal{N}| - 1$. In addition, the graph $T = (\mathcal{N}, \mathcal{E})$ must be connected and hence be a spanning tree. Suppose not, then $(\mathcal{N}, \mathcal{E})$ contains a cycle. Consider a node $j$ along with its predecessor $i$ and successor $k$ in the cycle. Without loss of generality, we assume $(i, j) \in \mathcal{E}$. If $(j, k) \in \mathcal{E}$, then by letting $y_{ij} = x_{ij} + \delta$ and $y_{jk} = x_{jk} + \delta$, the flow conservation constraint at node $j$ is not violated by the new flows. Similarly, if $(k, j) \in \mathcal{E}$, by letting $y_{ij} = x_{ij} + \delta$ and $y_{kj} = x_{jk} - \delta$, the flow conservation constraint at node $j$ is not violated by the new flows. Repeating such modification of flows, one can arrive at a new solution $\mathbf{y}$ that satisfies the flow conservation constraint at every node, which contradicts with the fact that $\mathbf{x}$ must be a unique solution. ■

## References

[BT97] D. Bertsimas and J.N. Tsitsiklis, *Introduction to Linear Optimization*, Springer, 1997.

[CCZ14] M. Conforti and G. Cornuéjols and G. Zambelli, *Integer Programming*, Springer, 2014.

**BDC6111: Introduction to Optimization** **Fall 2018, NUS**

# Lecture 9: Convex Analysis

*Lecturer: Zhenyu Hu*

## 9.1 Convex Sets

Recall that a set $C \subseteq \mathbb{R}^n$ is said to be convex if for any $0 \leq \theta \leq 1$, and $x, y \in C$, one has $\theta x + (1-\theta)y \in C$. Given a set $C$, the *convex hull* of $C$, denoted as $\text{conv}(C)$ is the set of all convex combinations of points in $C$:

$$\text{conv}(C) = \{\theta_1 x_1 + ... + \theta_k x_k | x_i \in C, \theta_i \geq 0, i = 1, ..., k, \theta_1 + ... + \theta_k = 1\}.$$

In the following, we describe some important example of convex sets.

### Convex cones

A set $C$ is called a *cone* if for every $x \in C$ and $\theta \geq 0$, we have $\theta x \in C$. A set $C$ is a *convex cone* if it is convex and a cone—for any $x_1, x_2 \in C$ and $\theta_1, \theta_2 \geq 0$, we have

$$\theta_1 x_1 + \theta_2 x_2 \in C.$$

A point of the form $\theta_1 x_1 + ... + \theta_k x_k$ is called a conic combination of $x_1, ..., x_k$. Similarly, a *conic hull* of a set $C$ is

$$\text{cone}(C) = \{\theta_1 x_1 + ... + \theta_k x_k | x_i \in C, \theta_i \geq 0, i = 1, ..., k\}.$$

Recall that the set $\{Ax \mid x \geq 0\}$ with $A$ being a matrix with dimension $m \times n$ defines a convex cone in $\mathbb{R}^m$. It is the conic hull of the vectors $A_1, ..., A_n \in \mathbb{R}^m$.

**Polyhedral cone:** A polyhedron of the form $C = \{x \in \mathbb{R}^n \mid Ax \geq 0\}$ is called a *polyhedral cone*. Clearly, it is a convex cone, and it can have at most one extreme point. In particular, if it does not contain a line, then it has a unique extreme point at zero. In this case, we call $C$ *pointed*. Any $x \neq 0$ cannot be extreme point since $3x/2 \in C, x/2 \in C$ and we have

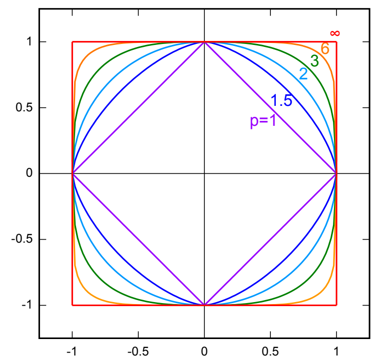
$$x = \frac{1}{2}\frac{3x}{2} + \frac{1}{2}\frac{x}{2}.$$

Similar to the definition of extreme point, a nonzero vector $x \in C$ is called an *extreme ray* if there are $n-1$ linearly independent constraints that are active at $x$. We will see below that a polyhedral cone is exactly the conic hull of all of its extreme rays.

**Norm cone:** Suppose $|| \cdot ||$ is any norm on $\mathbb{R}^n$. A *norm ball* with radius $r$ and center $x_c$ is defined as $B(x_c, r) = \{x \mid ||x - x_c|| \leq r\}$. The figure below shows the unit norm ball (actually its boundary) at the origin for p-norm with different p values.

The *norm cone* is defined as the set

$$C = \{(x, t) \mid ||x|| \leq t\} \subseteq \mathbb{R}^{n+1}.$$




One important special case is the *second-order cone* where the norm is specified by 2-norm (or Euclidean norm):

$$\begin{aligned} C =& \{(x,t) \in \mathbb{R}^{n+1} \mid ||x||_2 \leq t\} \\ =& \left\{ \begin{bmatrix} x \\ t \end{bmatrix} \middle| [x' \quad t] \begin{bmatrix} I & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ t \end{bmatrix} \leq 0, t \geq 0 \right\}. \end{aligned}$$

**Positive semidefinite cone:** Let $\mathbb{S}^n$ denote the set of all symmetric $n \times n$ matrices:

$$\mathbb{S}^n = \{X \in \mathbb{R}^{n\times n} \mid X = X'\}.$$

The positive semidefinite cone $\mathbb{S}^n_+$ is defined as

$$\mathbb{S}^n_+ = \{X \in \mathbb{S}^n \mid X \succeq 0\}.$$

Note that for any $\theta_1, \theta_2 \geq 0$ and $A, B \in \mathbb{S}^n_+$, we have $\theta_1 A + \theta_2 B \in \mathbb{S}^n_+$, and hence $\mathbb{S}^n_+$ is a convex cone. As an example, consider

$$\left\{ X \in \mathbb{S}^2 \middle| X = \begin{bmatrix} x & y \\ y & z \end{bmatrix} \succeq 0 \right\},$$

which is the same as

$$\{(x,y,z) \in \mathbb{R}^3 \mid x, z \geq 0, xz - y^2 \geq 0\}.$$

Alternatively, we can express $\mathbb{S}^n_+$ as

$$\mathbb{S}^n_+ = \bigcap_{z \in \mathbb{R}^n} \{X \in \mathbb{S}^n \mid z'Xz \geq 0\},$$

which is the intersection of infinite number of halfspaces and hence convex.

## Polyhedron

Here, we discuss an alternative description of the polyhedron $\{x \in \mathbb{R}^n | Ax \geq b\}$.

**Theorem 9.1 (Resolution Theorem)** *Let $P = \{x \in \mathbb{R}^n | Ax \geq b\}$ be a nonempty polyhedron with at least one extreme point. Let $v_1, ..., v_k$ be the extreme points, and let $w_1, ..., w_r$ be a complete set of extreme rays of the cone $\{x \in \mathbb{R}^n | Ax \geq 0\}$. Let*

$$Q = \left\{ \sum_{i=1}^{k} \lambda_i v_i + \sum_{j=1}^{r} \theta_j w_j \middle| \lambda_i \geq 0, \theta_j \geq 0, \sum_{i=1}^{k} \lambda_i = 1 \right\}.$$

*Then,* $Q = P$.

*Proof:* See the proof of Theorem 4.15 in [BT97]. ∎

The above theorem essentially says any polyhedron can represented as the convex hull of its extreme points plus the conic hull of its extreme rays.

**Example 9.2** *Consider the polyhedron defined by the constraints*

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

*One can find the extreme points as* $v_1 = (0,2), v_2 = (0,1), v_3 = (1,0)$*, and the extreme rays as* $w_1 = (1,1), w_2 = (1,0)$*. The vector* $y = (2,2)$*, for example, can be represented as* $y = v_2 + w_1 + w_2$*.*

## 9.2 Generalized Inequalities

A cone $K \subseteq \mathbb{R}^n$ is called a *proper cone* if it satisfies the following

- $K$ is convex.
- $K$ is closed, i.e., if $x_i \in K$ for any $i \geq 1$, and $\lim_{i\to\infty} x_i = x$, then $x \in K$.
- $K$ has nonempty interior, i.e., there exists $x \in K$ and $\epsilon > 0$ such that $\{y \mid ||y - x||_2 \leq \epsilon\} \subseteq K$.
- $K$ is pointed, i.e., it does not contain a line.

A proper cone $K$ can be used to define a *generalized inequality*, which is partial ordering defined as follows:

$$x \preceq_K y \Longleftrightarrow y - x \in K.$$

The corresponding strict partial ordering can be defined as:

$$x \prec_K y \Longleftrightarrow y - x \in \text{int } K,$$

where int $K$ denotes the interior of $K$.

The idea is to generalize the usual componentwise inequality "$\leq$" we used in linear programming problem.

**Nonnegative orthant:** Consider the special case when $K = \mathbb{R}^n_+$, the nonnegative orthant. It can be easily verified that it is a proper cone. In this case, $y - x \in \mathbb{R}^n_+$ is the same as $x \leq y$, i.e.,

$$x_i \leq y_i, i = 1, ..., n.$$

Similarly, the strict partial ordering $x \prec_{\mathbb{R}^n_+} y$ corresponds to $x_i < y_i, i = 1, ..., n$.

**Positive semidefinite cone:** From the representation

$$\mathbb{S}^n_+ = \bigcap_{z\in\mathbb{R}^n} \{X \in \mathbb{S}^n \mid z'Xz \geq 0\},$$

the positive semidefinite cone is the intersection of infinite number of closed half-spaces, and hence both closed and convex. In addition, if $X \in \mathbb{S}^n_+$ and $-X \in \mathbb{S}^n_+$, then $X = 0$, and hence $\mathbb{S}^n_+$ cannot contain a line.

To show that $\mathbb{S}^n_+$ has nonempty interior, we first generalize the inner-product and Euclidean norm defined for vectors to matrices. Given two matrices $X, Y \in \mathbb{R}^{m\times n}$, the inner product on $\mathbb{R}^{m\times n}$ is defined as

$$\langle X, Y\rangle = \mathrm{tr}(X'Y) = \sum_{i=1}^{m}\sum_{j=1}^{n} X_{ij}Y_{ij},$$

where $\mathrm{tr}(A)$ denotes the trace of the matrix $A$. The "Euclidean norm" or *Frobenius norm* of a matrix $X \in \mathbb{R}^{m\times n}$ is given by

$$\|X\|_F = (\langle X, X\rangle)^{1/2} = (\mathrm{tr}(X'X))^{1/2} = \sqrt{\left(\sum_{i=1}^{m}\sum_{j=1}^{n} X_{ij}^2\right)}.$$

We show that the identity matrix $I$ is in the interior of $\mathbb{S}^n_+$. Given $\epsilon > 0$, consider the norm ball

$$\{Y \mid \|Y - I\|_F \le \epsilon\}.$$

By definition $\left(\sum_{i=1}^{m}\sum_{j=1}^{n}(Y_{ij} - I_{ij})^2\right) \le \epsilon$, which implies

$$1 - \epsilon \le Y_{ii} \le 1 + \epsilon, \ \ -\epsilon \le Y_{ij} \le \epsilon, i \ne j.$$

It follows that

$$|Y_{ii}| \ge 1 - \epsilon \ge (n-1)\epsilon \ge \sum_{j\ne i} |Y_{ij}|$$

for $\epsilon$ small enough, which implies $Y \in \mathbb{S}^n_+$. In fact, any $X \succ 0$ is in the interior of $\mathbb{S}^n_+$, i.e., $\mathrm{int}\ \mathbb{S}^n_+ = \mathbb{S}^n_{++}$, the set of positive definite matrices.

In this case, $X \preceq Y$ means $Y - X \succeq 0$.

## Dual cones

Let $K$ be a cone. The set

$$K^* = \{y \mid x'y \ge 0 \text{ for all } x \in K\}$$

is called the *dual cone* of $K$. Clearly, for any $y_1, y_2 \in K^*$ and $\theta_1, \theta_2 \ge 0$, we have

$$x'(\theta_1 y_1 + \theta_2 y_2) = \theta_1 x'y_1 + \theta_2 x'y_2 \ge 0, \ \forall x \in K.$$

Hence, $K^*$ is a convex cone (regardless of whether $K$ is convex or not).

**Subspace:** The dual cone of a subspace $V \subseteq \mathbb{R}^n$ is its orthogonal complement $V^\perp = \{y \mid y'v = 0, \ \forall v \in V\}$.

**Nonnegative orthant:** The dual cone $\mathbb{R}^n_+$ is itself:

$$y'x \ge 0, \ \forall x \ge 0 \iff y \ge 0.$$

Such cone is called *self-dual*.

**Positive semidefinite cone:** The positive semi-definite cone $\mathbb{S}^n_+$ is self-dual on $\mathbb{S}^n$. Suppose $Y \in (\mathbb{S}^n_+)^*$, i.e.,

$$\langle X, Y\rangle = \mathrm{tr}(XY) \ge 0, \ \forall X \succeq 0.$$

We show that $Y \in \mathbb{S}^n_+$. If, on the contrary, $Y \notin \mathbb{S}^n$, then there exists $z \in \mathbb{R}^n$ such that

$$z'Yz < 0.$$

Note that

$$z'Yz = \mathrm{tr}(zz'Y) < 0$$

and the matrix $zz' \in \mathbb{S}^n_+$, which contradicts with $Y \in (\mathbb{S}^n)^*$. This shows $(\mathbb{S}^n_+)^* \subseteq \mathbb{S}^n_+$.

To show the other direction, suppose $X, Y \in \mathbb{S}^n_+$. By eigenvalue decomposition,

$$X = Q\Lambda Q',$$

where

$$\Lambda = \begin{bmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_n \end{bmatrix}$$

with $\lambda_i \geq 0$ being the eigenvalues of $X$ and $Q = [q_1 \; \dots \; q_n]$ with $q_i$ being the eigenvectors of $X$. It follows that

$$X = Q\Lambda Q' = \begin{bmatrix} q_1 & \dots & q_n \end{bmatrix} \begin{bmatrix} \lambda_1 q_1' \\ \vdots \\ \lambda_n q_n' \end{bmatrix} = \sum_{i=1}^{n} \lambda_i q_i q_i'.$$

Hence, we have

$$\mathrm{tr}(YX) = \mathrm{tr}\left(Y \sum_{i=1}^{n} \lambda_i q_i q_i'\right) = \sum_{i=1}^{n} \lambda_i \mathrm{tr}\left(q_i q_i' Y\right) = \sum_{i=1}^{n} \lambda_i q_i' Y q_i \geq 0,$$

i.e., $Y \in (\mathbb{S}^n_+)^*$.

**Norm cone:** The dual of the cone $K = \{(x,t) \in \mathbb{R}^{n+1} \mid \|x\| \leq t\}$ is the cone defined by the dual norm, i.e.,

$$K^* = \{(u,v) \in \mathbb{R}^{n+1} \mid \|u\|_* \leq v\},$$

where the dual norm is defined by $\|u\|_* = \sup\{u'x \mid \|x\| \leq 1\}$. To show this, suppose $(u,v) \in K^*$, i.e.,

$$u'x + vt \geq 0, \; \forall (x,t) \in K,$$

which is equivalent as

$$u'(-x/t) \leq v, \; \forall (x,t) \in K.$$

By $(x,t) \in K$, we have $\|x\| \leq t \iff \| -x/t\| \leq 1$. Hence,

$$\sup\{u'(-x/t) \mid (x,t) \in K\} = \sup\{u'(-x/t) \mid \| -x/t\| \leq 1\} = \|u\|_* \leq v.$$

Conversely, suppose $\|u\|_* \leq v$ and $\|x\| \leq t \iff \| -x/t\| \leq 1$, we have

$$u'(-x/t) \leq \|u\|_* \leq v,$$

which implies $u'x + vt \geq 0$.

As a special case, the dual norm of Euclidean norm is also Euclidean norm. To see this, consider

$$\sup\{u'x \mid \|x\|_2 \leq 1\}.$$

By Cauchy-Schwarz inequality, we have $u'x \leq \|z\|_2 \|x\|_2 = \|z\|_2$. Correspondingly, the second-order cone is self-dual.

More generally, the dual of $p$-norm is the $q$-norm, where $q$ satisfies $1/p + 1/q = 1$.

### Dual generalized inequalities

If $K$ is a proper cone, then so its dual $K^*$, and moreover $K^{**} = K$. The generalized inequality induced by $K^*$: $\preceq_{K^*}$ is referred to as the dual of the generalized inequality $\preceq_K$.

The generalized inequality and its dual induce the following important relationship:

- $x \preceq_K y$ if and only if $\lambda' x \le \lambda' y$ for all $\lambda \succeq_{K^*} 0$. This directly follows from the definition of dual cone: $y - x \in K$ if and only if $\lambda'(y - x) \ge 0$ for any $\lambda \in K^*$.
- $x \prec_K y$ if and only if $\lambda' x < \lambda' y$ for all $\lambda \succeq_{K^*} 0$, and $\lambda \ne 0$.

## 9.3 Separating Hyperplane Theorem

We first state the following version of separating hyperplane theorem under a stronger assumption. The distance between two sets $C$ and $D$ is defined as

$$\operatorname{dist}(C, D) = \inf\{\|u - v\|_2 \mid u \in C, v \in D\}.$$

**Theorem 9.3 (Strict Separating Hyperplane Theorem)** *Suppose $C$ and $D$ are two convex sets with* $\operatorname{dist}(C, D) > 0$, *and there exists $c \in C, d \in D$ such that $\|c - d\|_2 = \operatorname{dist}(C, D)$. Then there exist $a \ne 0$ and $b$ such that $a'x < b$ for all $x \in C$ and $a'x > b$ for all $x \in D$. The hyperplane $\{x \mid a'x = b\}$ is called a* separating hyperplane *for the sets $C$ and $D$.*

*Proof*: Let

$$a = d - c, \;\; b = \frac{\|d\|_2^2 - \|c\|_2^2}{2}.$$

Consider the hyperplane

$$f(x) = a'x - b = (d - c)'x - \frac{\|d\|_2^2 - \|c\|_2^2}{2} = (d - c)'(x - (d + c)/2).$$

We show that $f$ is positive on $D$; the proof that $f$ is negative on $C$ is similar. Suppose, on the contrary, there exists $u \in D$ such that

$$f(u) = (d - c)'(u - (d + c)/2) \le 0.$$

Note that

$$f(u) = (d - c)'(u - (d + c)/2) = (d - c)'(u - d + d - (d + c)/2) = (d - c)'(u - d) + \|d - c\|_2^2/2.$$

Hence, $f(u) \le 0$, implies $(d - c)'(u - d) < 0$. Now consider the line segment $\theta u + (1 - \theta)d$, for $\theta \in [0, 1]$. Observe that

$$\frac{d}{d\theta}\|d + \theta(u - d) - c\|_2^2\Big|_{\theta=0} = 2(d - c)'(u - d) < 0.$$

That is, for small enough $\theta > 0$, we have

$$\|d + \theta(u - d) - c\|_2 < \|d - c\|_2,$$

which contradicts with the fact that $d$ is the closest point to $c$ in $D$. ■

As a special case, when $D$ is a closed convex set and $C = \{x_0\}$ with $x_0 \notin D$, there exists $\epsilon > 0$ such that the (Euclidean) norm ball $B(x_0, \epsilon)$ does not intersect with $D$, i.e., $B(x_0, \epsilon) \cap D = \emptyset$. This implies $\operatorname{dist}(C, D) > 0$.

In addition, since $D$ is closed, there exists $d \in D$ such that $\|x_0 - d\|_2 = \text{dist}(C, D)$. By Theorem 9.3, there exists $a \neq 0, b$ such that $a'x_0 < b < a'x$ for all $x \in D$.

Recall the Farkas' Lemma introduced in Lecture 4. We now provide an alternative proof based on the separating hyperplane theorem.

**Theorem 9.4 (Farkas' Lemma)** *Let $a_1, ..., a_m$ and $c$ be vectors in $\mathbb{R}^n$. Then, exactly one of the following alternatives holds:*

1. *There exists some $p \geq 0$ in $\mathbb{R}^m$ such that*

$$\sum_{i=1}^{m} p_i a_i = c.$$

2. *There exists some vector $d \in \mathbb{R}^n$, such that*

$$Ad = \begin{bmatrix} a'_1 \\ \vdots \\ a'_m \end{bmatrix} d \geq 0$$

*and $c'd < 0$.*

*Proof:* Clearly, when the first alternative holds, for any $d \in \mathbb{R}^n$ with $Ad \geq 0$, we have

$$p'Ad = c'd \geq 0,$$

and the second alternative cannot hold.

The nontrivial part is when the first alternative does not hold, i.e., we cannot find a vector $p \geq 0$ in $\mathbb{R}^m$ such that

$$\sum_{i=1}^{m} p_i a_i = c.$$

In this case, the polyhedron cone generated by the conic hull of $a_1, ..., a_m$: $D = \{\sum_{i=1}^m p_i a_i \mid p \geq 0\}$. When the first alternative is invalid, we know $c \notin D$. By Theorem 9.3, there exists $d \neq 0$ such that $d'c < d'x$ for all $x \in D$. In particular, since $x = 0 \in D$, we have $d'c < 0$. On the other hand, for any $\theta > 0$, we have $\theta a_i \in D$ and $\theta d'a_i > d'c$ or equivalently $d'a_i > d'c/\theta$. It follows that

$$d'a_i \geq \lim_{\theta \to \infty} \frac{d'c}{\theta} = 0, \ \forall i = 1, ..., m.$$

That is, $Ad \geq 0$ and $c'd < 0$. ∎

Using Theorem 9.3, we can further establish the following separating hyperplane theorem under weaker condition.

**Theorem 9.5 (Separating Hyperplane Theorem)** *Suppose $C$ and $D$ are two convex sets with $C \cap D = \emptyset$. Then there exist $a \neq 0$ and $b$ such that $a'x \leq b$ for all $x \in C$ and $a'y \geq b$ for all $y \in D$.*

*Proof:* Consider the set

$$S = \{y - x \mid x \in C, y \in D\}.$$

By $C \cap D = \emptyset$, we have $0 \notin S$ and it can be easily verified that $S$ is convex. The closure of set $S$, denoted as $\text{cl}(S)$, is the set of all limit points of convergent sequences in $S$. By definition, $S \subseteq \text{cl}(S)$ and $\text{cl}(S)$ is closed (and convex if $S$ is convex).

If $0 \notin \text{cl}(S)$, then by Theorem 9.3, there exist $a \neq 0, b$ such that $a'0 < b < a'd$ for all $d \in \text{cl}(S)$. This implies $a'(y-x) > 0$ for any $x \in C, y \in D$ and hence $a'x \leq b \leq a'y$ for some $b$ and for all $x \in C, y \in D$.

Suppose now that $0 \in \text{cl}(S)$. If $S$ has empty interior, then it is contained in some hyperplane $\{z \mid a'z = b\}$ and since $0 \in \text{cl}(S)$, $b = 0$. As a result, for any $d \in S$, $a'd = 0$, i.e., $a'y = a'x$ for any $x \in C, y \in D$, which is a trivial separating hyperplane.

If $S$ has nonempty interior, for any $\epsilon > 0$, consider the set

$$S_{-\epsilon} = \{d \mid B(d, \epsilon) \subseteq S\}.$$

That is, $S_{-\epsilon}$ is the collection of points that have their $\epsilon$-ball contained in $S$. It can be shown that $S_{-\epsilon}$ is convex given that $S$ is convex and hence its closure $\text{cl}(S_{-\epsilon})$ is closed and convex. In addition, $0 \notin \text{cl}(S_{-\epsilon})$ and hence by Theorem 9.3, there exists $a(\epsilon)$ such that

$$a(\epsilon)'d > 0, \ \forall d \in S_{-\epsilon}.$$

We can always normalize $a(\epsilon)$ such that $\|a(\epsilon)\|_2 = 1$. Now consider a decreasing sequence $\epsilon_n, n = 1, 2, ...$ with $\lim_{n\to\infty} \epsilon_n = 0$. Correspondingly, we have $a(\epsilon_n)$ with $\|a(\epsilon_n)\|_2 = 1$ and

$$a(\epsilon_n)'d > 0, \ \forall d \in S_{-\epsilon_n}.$$

Since $a(\epsilon_n)$ is a bounded sequence, there exists convergent subsequence, i.e., $\lim_{k\to\infty} a(\epsilon_{n_k}) \to \bar{a}$. For any $d \in \text{int}(S)$, there exists $\epsilon_{n_i}$ such that $d \in S_{\epsilon_{n_i}}$ and

$$a'_{\epsilon_{n_j}} d > 0, \ \forall j \geq i.$$

Hence, $\bar{a}'d \geq 0$ for any $d \in \text{int}(S)$. The inequality is preserved by taking limits under all convergent sequence in $\text{int}(S)$ and we have $\bar{a}'d \geq 0$ for any $d \in S$, i.e., $\bar{a}'x \geq \bar{a}'y$ for all $x \in C, y \in D$. ∎

A simple corollary of Theorem 9.5 is the so-called *supporting hyperplane theorem.* Let $C$ be convex set (assume that $\text{int}(C) \neq \emptyset$) and $x_0$ be a point on its boundary, i.e., $x_0 \in \text{cl}(C)$, $x_0 \notin \text{int}(C)$. Clearly, $\text{int}(C) \cap \{x_0\} = \emptyset$. By Theorem 9.5, there exists $a \neq 0$ such that $a'x_0 \geq a'x$ for any $x \in \text{int}(C)$, which also implies $a'x_0 \geq a'x$ for any $x \in C$. The hyperplane $\{x \mid a'x = a'x_0\}$ is called the *supporting hyperplane* of $C$ at $x_0$.

## 9.4 Convex Functions

Recall that a function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if its domain $\text{dom}f$ is convex and for all $x, y \in \text{dom}f$ and $\theta \in [0, 1]$, we have

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y).$$

Below are several equivalent characterizations:

**First-Order:** A differential function $f$ is convex if and only if for any $x, y \in \text{dom}f$

$$f(x) + \nabla f(x)'(y - x) \leq f(y).$$

**Second-Order:** A twice-differential function $f$ is convex if and only if for any $x \in \mathrm{dom} f$

$$\nabla^2 f(x) \succeq 0.$$

**Epigraph:** The graph of a function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$\{(x, f(x)) \mid x \in \mathrm{dom} f\}.$$

The *epigraph* is defined as

$$\mathrm{epi} f = \{(x, t) \mid x \in \mathrm{dom} f, f(x) \leq t\}.$$

The function $f$ is convex if and only if its epigraph $\mathrm{epi} f$ is a convex set.

## Convexity with respect to generalized inequalities

Suppose $K \subseteq \mathbb{R}^m$ is a proper cone with associated generalized inequality $\preceq_K$. A mapping $f : \mathbb{R}^n \to \mathbb{R}^m$ that maps a vector $x \in \mathbb{R}^n$ to a vector $(f_1(x), ..., f_m(x)) \in \mathbb{R}^m$ is $K$-convex if for all $x, y$ and $0 \leq \theta \leq 1$,

$$f(\theta x + (1-\theta)y) \preceq_K \theta f(x) + (1-\theta) f(y).$$

**Nonnegative orthant:** For $K = \mathbb{R}^m_+$, the mapping $f$ is $\mathbb{R}^m_+$-convex is the same as requiring the function $f_i(x)$ to be convex for all $i = 1, ..., m$.

**Positive semi-definite cone:** In this case, $K = \mathbb{S}^m_+$, and $f : \mathbb{R}^n \to \mathbb{S}^m$ is a mapping from vectors to symmetric matrices. The $\mathbb{S}^m_+$-convexity is also called *matrix convexity*. To show $f$ is $\mathbb{S}^m_+$-convex, it is equivalent to show that $z' f(x) z$ is a convex function in $x$ for any $z \in \mathbb{R}^m$.

- $f(X) = XX'$, where $X \in \mathbb{R}^{n \times m}$.
- $f(X) = X^{-1}$, where $X \in \mathbb{S}^n_{++}$.

# References

[BT97] D. Bertsimas and J.N. Tsitsiklis, *Introduction to Linear Optimization*, Springer, 1997.

[BV03] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2003.

| BDC6111: Introduction to Optimization | Fall 2018, NUS |
|---|---|

# Lecture 10: Convex Programs

*Lecturer: Zhenyu Hu*

We have specified in the first lecture a general convex programming problem of the form

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_i(x) \le 0, i = 1, ..., l, \\ & a_i'x = b_i, i = 1, ..., m, \end{array} \tag{10.1}$$

where $f(\cdot), g_1(\cdot), ..., g_l(\cdot)$ are convex functions. The goal in this lecture is to introduce several important special cases of it.

## 10.1 Quadratic Optimization

### Quadratic program (QP):

$$\begin{aligned} \min\ & x'Px + q'x + r \\ \text{s.t.}\ & Gx \le h, \\ & Ax = b, \end{aligned}$$

where $P \in \mathbb{S}_+^n, G \in \mathbb{R}^{l\times n}, A \in \mathbb{R}^{m\times n}$. When $P = 0$, the quadratic program reduces to linear program.

**Markowitz portfolio optimization problem:** One classic QP is the Markowitz portfolio optimization problem. Consider $n$ assets or stocks. Let $\tilde{r}_i$ be the random return of stock $i$ and $x_i$ be the proportion of budget invested in stock $i$, $i = 1, ..., n$. Suppose $(\tilde{r}_1, ..., \tilde{r}_n)$ has mean $(r_1, ..., r_n)$ and covariance $\Sigma$. It follows that the return from the portfolio $\sum_{i=1}^n \tilde{r}_i x_i$ has

$$\mathbb{E}[\sum_{i=1}^n \tilde{r}_i x_i] = \sum_{i=1}^n r_i x_i,\ \text{var}\left(\sum_{i=1}^n \tilde{r}_i x_i\right) = x'\Sigma x.$$

One seeks to find a portfolio $x$ that gives at least an expected return of $v_{\min}$ and minimizes the risk:

$$\begin{aligned} \min\ & x'\Sigma x \\ \text{s.t.}\ & r'x \ge v_{\min}, \\ & 1'x = 1, \\ & x \ge 0. \end{aligned}$$

### Quadratically constrained quadratic program (QCQP):

$$\begin{aligned} \min\ & x'P_0x + q_0'x + r_0 \\ \text{s.t.}\ & x'P_ix + q_i'x + r_i \le 0, i = 1, ..., l, \\ & Ax = b, \end{aligned}$$

where $P_i \in \mathbb{S}_+^n, i = 0, 1, ..., l.$[1]

[1] Some literature would refer to QP and QCQP introduced here as convex QP and convex QCQP, with QP and QCQP referring to the more general case where $P$ or $P_i$ not being required to be positive semi-definite.

For the portfolio optimization problem, if one seeks to maximize the expected return with a maximum bearable risk $\sigma_{\max}$, then the problem is formulated as QCQP:

$$\begin{aligned} \max\ & r'x \\ \text{s.t. } & x'\Sigma x \le \sigma_{\max}, \\ & 1'x = 1, \\ & x \ge 0. \end{aligned}$$

## Second-order cone programming (SOCP):

$$\begin{aligned} \min\ & f'x \\ \text{s.t. } & \|A_i x + b_i\|_2 \le c_i'x + d_i, i = 1, ..., l, \\ & Fx = g, \end{aligned}$$

where $A_i \in \mathbb{R}^{n_i \times n}, F \in \mathbb{R}^{m \times n}$. Note that when $c_i = 0$, the above SOCP reduces to a QCQP: by squaring both sides of the constraint $\|A_i x + b_i\|_2 \le d_i$ (note that one must have $d_i \ge 0$), one has

$$x'A_i'A_i x + 2b_i'A_i x + b_i'b_i - d_i^2 \le 0.$$

On the other hand, every QCQP can be formulated as an SOCP. First note that QCQP can always be reformulated as one that has linear objective:

$$\begin{aligned} \min\ & t \\ \text{s.t. } & \begin{bmatrix} x' & t \end{bmatrix} \begin{bmatrix} P_0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ t \end{bmatrix} + \begin{bmatrix} q_0' & -1 \end{bmatrix} \begin{bmatrix} x \\ t \end{bmatrix} + r_0 \le 0, \\ & \begin{bmatrix} x' & t \end{bmatrix} \begin{bmatrix} P_i & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ t \end{bmatrix} + \begin{bmatrix} q_i' & 0 \end{bmatrix} \begin{bmatrix} x \\ t \end{bmatrix} + r_i \le 0,\ i = 1, ..., l, \\ & \begin{bmatrix} A & 0 \end{bmatrix} \begin{bmatrix} x \\ t \end{bmatrix} = b. \end{aligned}$$

It is then sufficient to show that the quadratic constraint of the form $x'Px + q'x + r \le 0$ can be reformulated as a second-order cone constraint. Since $P \in \mathbb{S}_+^n$, by Cholesky factorization we have

$$P = A'A,$$

where $A$ is some nonsingular lower triangular matrix. Note that

$$x'Px + q'x + r = x'A'Ax + \frac{(1 + q'x + r)^2}{4} - \frac{(1 - q'x - r)^2}{4} \le 0,$$

which is equivalent to

$$\left\| \begin{bmatrix} (1 + q'x + r)/2 \\ Ax \end{bmatrix} \right\|_2 \le \frac{1 - q'x - r}{2}.$$

We remark when $c_i \ne 0$, there are some SOCP problems that cannot be formulated as QCQP (see homework),[2] and hence SOCP is a more general class of problems than QCQP.

[2] More strictly speaking, second-order cone constraint can be reformulated as constraint of the form $x'Px + q'x + r \le 0$ plus a linear constraint but the matrix $P$ may not be positive semi-definite.

**Robust optimization:** As an application, consider the robust optimization problem studied in Lecture 6, where we considered the constraint

$$\sum_{j=1}^{n}(a_{ij}+\delta_{ij}z_{ij})x_j \le b_i,$$

to be satisfied for all $z_i=(z_{i1},...,z_{in})$ in the uncertainty set

$$\left\{ z_i \,\middle|\, \sum_{j=1}^{n}|z_{ij}| \le \alpha_i n, -1 \le z_{ij} \le 1 \right\}.$$

Instead of using $l_1$-norm, we can also $l_2$-norm to exclude the extreme values of $z_{ij}$ by considering the uncertainty set:

$$\left\{ z_i \,\middle|\, \|z_i\|_2 \le \alpha_i\sqrt{n}, -1 \le z_{ij} \le 1 \right\}.$$

The problem of finding the worst case realization is then an SOCP:

$$\begin{aligned} g_i(x) = \sum_{j=1}^{n} a_{ij}x_j + \max_{z_i} \quad & \sum_{j=1}^{n}\delta_{ij}x_j z_{ij} \\ s.t. \quad & \|z_i\|_2 \le \alpha_i\sqrt{n} \\ & -1 \le z_{ij} \le 1. \end{aligned}$$

One good further reading for more applications of SOCP is [Alizadeh and Goldfarb 2003].

## 10.2 Conic Form Problems

The form (10.1) can be made even more general by using generalized inequalities:

$$\begin{aligned} \min \ & f_0(x) \\ \text{s.t. } & f_i(x) \preceq_{K_i} 0, i=1,...,l, \\ & Ax = b, \end{aligned}$$

where $f_0 : \mathbb{R}^n \to \mathbb{R}$ is convex, $K_i \subseteq \mathbb{R}^{k_i}$ are proper cones and $f_i : \mathbb{R}^n \to \mathbb{R}^{k_i}$ are $K_i$-convex.

The simplest special case of the above form is when $l=1$ and $f_0, f_1$ are affine function and affine mapping ($f_1$ is clearly $K$-convex for any cone $K$):

$$\begin{aligned} \min \ & c'x \\ \text{s.t. } & Fx + g \preceq_K 0, \\ & Ax = b. \end{aligned}$$

Problems of the above form are called *conic form problems* or *conic programs.* Clearly, when $K$ is the nonnegative orthant, the above problem is a linear program. Similar to LP, conic form problem of the following form

$$\begin{aligned} \min \ & c'x \\ \text{s.t. } & Ax = b, \\ & x \succeq_K 0, \end{aligned}$$

is referred to as a *conic form problem in standard form.*

## Second-order cone programming (SOCP):

We have already seen that second-order cone problems are quadratic optimization problems. In fact, as the name suggest, they can be expressed as conic form problems as follows:

$$\begin{aligned} \min\ & f'x \\ \text{s.t.}\ & -(A_i x + b_i, c_i' x + d_i) \preceq_{K_i} 0, i = 1, ..., l, \\ & Fx = g, \end{aligned}$$

where

$$K_i = \left\{ (y, t) \in \mathbb{R}^{n_i+1} \mid \|y\|_2 \leq t \right\},$$

is the second-order cone in $\mathbb{R}^{n_i+1}$. Note that we can always introduce new variables $y_i = A_i x + b_i, t_i = c_i' x + d_i$ such that requiring $(y_i, t_i) \in K_i$ is the same as requiring $(y_1, t_1, y_2, t_2, ..., y_l, t_l) \in K$, where $K = K_1 \times K_2 \times ... \times K_l$, which is also a cone.

## Semidefinite programming (SDP):

When $K = \mathbb{S}_+^k$, the associated conic form problem is called a *semidefinite program*, and has the form

$$\begin{aligned} \min\ & c'x \\ \text{s.t.}\ & x_1 F_1 + x_2 F_2 + ... + x_n F_n + G \preceq 0, \\ & Ax = b, \end{aligned}$$

where $F_1, ..., F_n, G \in \mathbb{S}^k$ and $A \in \mathbb{R}^{m \times n}$. The inequality is also called *linear matrix inequality* (LMI). If the matrices $F_1, ..., F_n, G$ are all diagonal, then the LMI reduces to $k$ linear inequalities, and the SDP becomes an LP.

It is often useful to think of $x$ as a matrix instead of a vector. We use $X \in \mathbb{S}^n$ to denote the decision variables that are arranged as a matrix. The standard form SDP is:

$$\begin{aligned} \min\ & \mathrm{tr}(CX) \\ \text{s.t.}\ & \mathrm{tr}(A_i X) = b_i, i = 1, ..., m \\ & X \succeq 0, \end{aligned}$$

where $C, A_1, ..., A_m \in \mathbb{S}^n$. Note that

$$\mathrm{tr}(CX) = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij}$$

is a linear function in $X_{ij}$.

**Cartesian product of semidefinite cones:** Unlike LP, problems of the form, say,

$$\begin{aligned} \min\ & \mathrm{tr}(C_1 X_1) + \mathrm{tr}(C_2 X_2) \\ \text{s.t.}\ & \mathrm{tr}(A_i X_1) = b_i, i = 1, ..., m, \\ & \mathrm{tr}(B_j X_2) = d_j, j = 1, ..., l, \\ & X_1 \succeq 0, X_2 \succeq 0 \end{aligned}$$

is not—strictly speaking—in standard form. One can, however, transform it to standard form by defining

$$X = \begin{bmatrix} X_1 & Y \\ Y' & X_2 \end{bmatrix}$$

and introducing the constraints $Y = 0$. Note that the constraint $X_{ij} = 0$ (in our case the index $(i, j)$ such that it falls in the block matrix $Y$) can be expressed as $\mathrm{tr}(E_{ij}X) = 0$, where $E_{ij}$ is a symmetric matrix with 1 at entry $(i, j)$ and $(j, i)$, and zero elsewhere. The observation here is that, $X_1 \succeq 0, X_2 \succeq 0$ if and only if $X \succeq 0$ since

$$\begin{bmatrix} x' & y' \end{bmatrix} \begin{bmatrix} X_1 & 0 \\ 0' & X_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = x'X_1x + y'X_2y.$$

Similarly, consider the problem of the form

$$\begin{aligned} \min\ & c'x + \mathrm{tr}(CX) \\ \text{s.t. } & Ax = b \\ & \mathrm{tr}(B_jX) = d_j, j = 1, ..., l, \\ & x \geq 0, X \succeq 0, \end{aligned}$$

where we have a Cartesian product of a nonnegative orthant cone and a semidefinite cone. Note that nonnegative orthan cone can be viewed as the Cartesian product of $n$ one-dimensional semidefinite cone. Hence, we can transform the problem to standard form SDP via

$$\tilde{X} = \begin{bmatrix} \mathrm{diag}(x) & 0 \\ 0' & X \end{bmatrix},$$

where

$$\mathrm{diag}(x) = \begin{bmatrix} x_1 & 0 & \dots & 0 \\ 0 & x_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & x_n \end{bmatrix}.$$

**SOCP:** We first observe that a vector $(x, t) \in \mathbb{R}^n$ is in the second-order cone, i.e., $\|x\|_2 \leq t$ if and only if the matrix

$$X = \begin{bmatrix} tI & x \\ x' & t \end{bmatrix} \succeq 0.$$

To see this, note that the Schur complement[3] of $tI$ when $t > 0$ is

$$t - x'(tI)^{-1}x = t - \frac{1}{t}x'x.$$

- If $t = 0$, $X \succeq 0$ if and only if $x = 0$.
- If $t > 0$, $X \succeq 0$ if and only if $t - \frac{1}{t}x'x \geq 0$, which is equivalent to $\|x\|_2 \leq t$.

As a result, an SOCP constraint $\|x\|_2 \leq t$ can be reformulated as

$$X \succeq 0,\ \mathrm{tr}(E_{ii}X) = \mathrm{tr}(E_{n+1,n+1}X),\ i = 1, ..., n,\ \mathrm{tr}(E_{ij}X) = 0, i \neq j, i, j \neq n + 1.$$

**Moment problems:** Consider the following problem of bounding the tail probability of a random variable $\mathbb{P}(X \geq a)$ with first two moments information $\mathbb{E}[X] = M_1, \mathbb{E}[X^2] = M_2$. Let $\mu$ be a distribution function.

[3] Consider a matrix $X \in \mathbb{S}^n$ partitioned as

$$X = \begin{bmatrix} A & B \\ B' & C \end{bmatrix}.$$

If $\det(A) \neq 0$, the Schur complement of $A$ is defined as $C - B'A^{-1}B$. Similarly, if $\det(C) \neq 0$, the Schur complement of $C$ is $A - BC^{-1}B'$.

The problem can be formulated as:

$$\begin{aligned} \max_{\mu} \ & \int_{-\infty}^{+\infty} 1_{\{x \geq a\}} d\mu \\ \text{s.t.} \ & \int_{-\infty}^{+\infty} 1 d\mu = 1, \\ & \int_{-\infty}^{+\infty} x d\mu = M_1, \\ & \int_{-\infty}^{+\infty} x^2 d\mu = M_2, \end{aligned}$$

where $1_A$ is the indicator function of the set $A$. The above problem is an example of a *semi-infinite programming* problem. Similar to LP, the dual of the above problem (think about the case when $X$ is a discrete random variable) is

$$\begin{aligned} \min_{y_0, y_1, y_2} \ & y_0 + M_1 y_1 + M_2 y_2 \\ \text{s.t.} \ & y_0 + x y_1 + x^2 y_2 \geq 1, \ \forall x \geq a, \\ & y_0 + x y_1 + x^2 y_2 \geq 0, \ \forall x \in \mathbb{R}. \end{aligned}$$

We utilize the following important result in nonnegative polynomials: a polynomial $g(x) = \sum_{r=1}^{2k} y_r x^r \geq 0$ for all $x \in \mathbb{R}$ if and only a matrix $X \in \mathbb{S}^{k+1}$, such that

$$\sum_{i,j: i+j=r} X_{ij} = y_r, \ r = 0, ..., 2k, \ X \succeq 0.$$

It follows that the set of constraints $y_0 + x y_1 + x^2 y_2 \geq 0, \ \forall x \in \mathbb{R}$ is equivalent to

$$\begin{bmatrix} y_0 & y_1/2 \\ y_1/2 & y_2 \end{bmatrix} \succeq 0.$$

In addition, by letting $x = a + t^2$, the set of constraints $y_0 + x y_1 + x^2 y_2 \geq 1, \ \forall x \geq a$ is equivalent to

$$y_0 - 1 + (a + t^2) y_1 + (a + t^2)^2 y_2 = (y_0 - 1 + y_1 a + y_2 a^2) + (y_1 + 2a y_2) t^2 + y_2 t^4 \geq 0, \ \forall t \in \mathbb{R},$$

which is then reformulated as

$$\begin{bmatrix} y_0 - 1 + a y_1 + a^2 y_2 & 0 & x_{02} \\ 0 & x_{11} & 0 \\ x_{20} & 2 & y_2 \end{bmatrix} \succeq 0, \ \text{and} \ x_{02} + x_{11} + x_{20} = y_1 + 2a y_2.$$

More generally, the problem of finding an upper bound on tail probability with $k$ moments information can also be formulated as an SDP. Interested students are referred to [Bertsimas and Popescu 2005].

# References

[Alizadeh and Goldfarb 2003] F. Alizadeh and D. Goldfarb, 2003, "Second-order cone programming", *Mathematical Programming*, 95(1), pp. 3–51.

[Bertsimas and Popescu 2005] D. Bertsimas and I. Popescu, 2005, "Optimal inequalities in probability theory: a convex optimization approach", *SIAM Journal on Optimization*, 15(3), pp. 780–804.

[BV03] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2003.

**BDC6111: Introduction to Optimization** **Fall 2018, NUS**

# Lecture 11: Convex Duality

*Lecturer: Zhenyu Hu*

## 11.1 The Lagrange Dual Problem

We consider an optimization problem of the form

$$\begin{array}{ll} \min & f_0(x) \\ \text{s.t.} & f_i(x) \leq 0, i = 1, ..., l \\ & h_i(x) = 0, i = 1, ..., m. \\ & x \in \mathbb{R}^n. \end{array} \tag{11.1}$$

We assume the domain $\mathcal{D} = \bigcap_{i=1}^{l} \text{dom} f_i \cap \bigcap_{i=1}^{m} \text{dom} h_i$ is nonempty and denote the optimal value of (11.1) by $p^*$.

As in Lecture 4, we can define the Lagrangian as

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{l} \lambda_i f_i(x) + \sum_{i=1}^{m} \nu_i h_i(x),$$

with $\lambda_i \geq 0, i = 1, ..., l$.

The dual function is defined as

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu).$$

Note that $g(\lambda, \nu)$ is always concave even when (11.1) is not convex. In addition, we must have $g(\lambda, \nu) \leq p^*$ for $\lambda \geq 0$ since for any feasible solution $\tilde{x}$ to (11.1), we have

$$g(\lambda, \nu) \leq L(\tilde{x}, \lambda, \nu) \leq f_0(\tilde{x}).$$

The Lagrange dual problem is then defined as

$$\begin{array}{ll} \max & g(\lambda, \nu) \\ \text{s.t.} & \lambda \geq 0. \end{array} \tag{11.2}$$

We refer to a pair $(\lambda, \nu)$ with $\lambda \geq 0$ and $(\lambda, \nu) \in \text{dom} g$ as *dual feasible* and denote the optimal value of (11.2) by $d^*$.

The inequality $g(\lambda, \nu) \leq p^*$ immediately implies the weak duality:

$$d^* = \max_{\lambda \geq 0, \nu} g(\lambda, \nu) \leq p^*,$$

and the difference $p^* - d^*$ is referred to as the *duality gap*.

**Example 11.1 (Two-Way Partitioning Problem)** *Consider the problem*

$$\begin{array}{ll} \min & x'Wx \\ s.t. & x_i^2 = 1, i = 1, ..., n, \end{array}$$

*with $W \in \mathbb{S}^n$. It's called two-way partitioning because a feasible $x$ corresponds to a partition*

$$\{1, ..., n\} = \{i | x_i = 1\} \cup \{i | x_i = -1\}.$$

*If $i, j$ are in the same partition, it incurs a cost $W_{ij}$; if they are in different partitions, it incurs a cost $-W_{ij}$.*

*The Lagrangian for this problem can be written as*

$$\begin{aligned} L(x, \nu) &= x'Wx + \sum_{i=1}^{n} \nu_i (x_i^2 - 1) \\ &= x'(W + diag(\nu))x - 1'\nu. \end{aligned}$$

*The dual function is then*

$$g(\nu) = \begin{cases} -1'\nu, & \text{if} \quad W + diag(\nu) \succeq 0, \\ -\infty, & \text{otherwise.} \end{cases}$$

*We can construct lower bound for the primal problem by, for example, taking $\nu = -\lambda_{min}(W)1$. In this case, we must have*

$$p^* \geq n\lambda_{min}(W).$$

*The dual problem in this case is*

$$\begin{aligned} \max \quad & -1'\nu \\ s.t. \quad & W + diag(\nu) \succeq 0, \end{aligned}$$

*which is an SDP and can be solved much more efficiently compared to the primal.*

## 11.2 Duality Theory

Strong duality does not hold in general for problem (11.1). But if (11.1) is convex, i.e., of the form

$$\begin{aligned} \min \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, i = 1, ..., l \\ & Ax = b. \\ & x \in \mathbb{R}^n, \end{aligned} \tag{11.3}$$

with $A \in \mathbb{R}^{m \times n}$ and $f_0, ..., f_l$ being convex, then under certain conditions, we have strong duality.

One such condition is called *Slater's condition*: there exists an $x \in \text{relint}\mathcal{D}$ such that

$$f_i(x) < 0, i = 1, ..., l, \;\; Ax = b.$$

**Theorem 11.2 (Strong Duality)** *Suppose problem* (11.3) *has an optimal value $p^*$ and Slater's condition holds. Then there exists a dual feasible $(\lambda^*, \nu^*)$ with $g(\lambda^*, \nu^*) = d^* = p^*$.*

### Geometric Interpretation

Consider the general problem (11.1) and define

$$\mathcal{G} = \{(f_1(x), ..., f_l(x), h_1(x), ..., h_m(x), f_0(x)) \in \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R} | x \in \mathcal{D}\}.$$

Note that when (11.1) is of the form

$$\begin{array}{ll} \min & f_0(x) \\ \text{s.t.} & x \leq 0, \end{array}$$

then $\mathcal{G}$ is simply the graph of the objective function $f_0(x)$. We further define

$$\mathcal{A} = \{(u, v, t) \in \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R} | \exists x \in \mathcal{D}, f_i(x) \leq u_i, i = 1, ..., l, h_i(x) = v_i, i = 1, ..., m, f_0(x) \leq t\}.$$

Equivalently, we have $\mathcal{A} = \mathcal{G} + (\mathbb{R}^l_+ \times \{0\} \times \mathbb{R}_+)$.

We can express the optimal value of the primal as

$$p^* = \inf\{t | (0, 0, t) \in \mathcal{A}\}.$$

The dual function can be expressed as

$$g(\lambda, \nu) = \inf\{(\lambda, \nu, 1)'(u, v, t) | (u, v, t) \in \mathcal{A}\}$$

for $\lambda \geq 0$. In particular, if the infimum is attained, then $(\lambda, \nu, 1)'(u, v, t) \geq (\lambda, \nu, 1)'(u^*, v^*, t^*)$, and the hyperplane

$$\{(u, v, t) | (\lambda, \nu, 1)'(u, v, t) = (\lambda, \nu, 1)'(u^*, v^*, t^*)\}$$

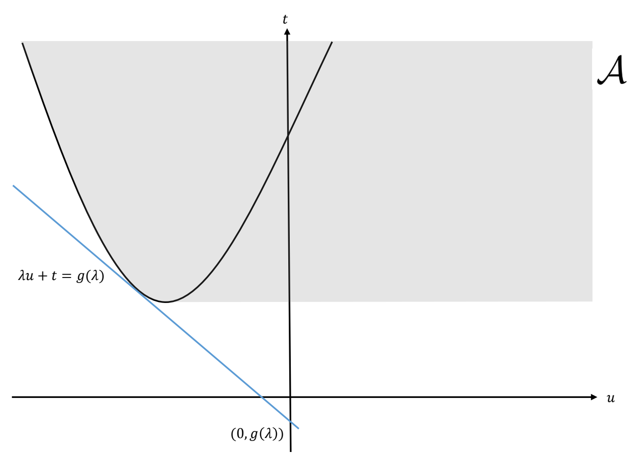
defines a supporting hyperplane to $\mathcal{A}$ at $(u^*, v^*, t^*)$.

Strong duality holds if and only if there exists a nonvertical supporting hyperplane to $\mathcal{A}$ at $(0, 0, p^*)$.

**Example 11.3** *Consider the one-dimensional problem*

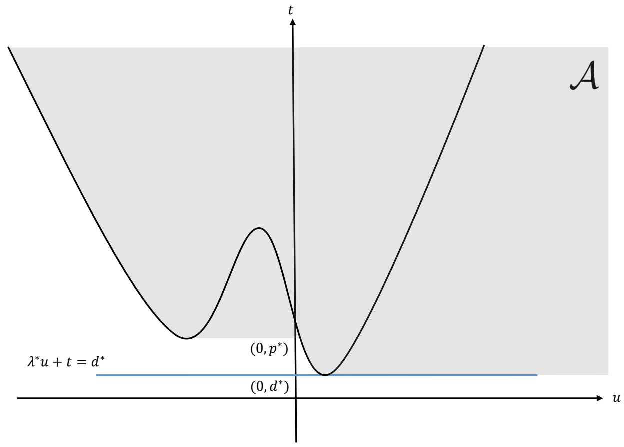
$$\begin{array}{ll} \min & f_0(x) \\ \text{s.t.} & x \leq 0, \end{array}$$

*with* $f_0(x) = x^2 + 2x + 2$.




*In this case,* $\mathcal{G} = \{(u, t) | t = u^2 + 2u + 2\}$ *and* $\mathcal{A}$ *is illustrated as the shaded region in the figure above.*

*If, on the other hand,* $f_0(x)$ *is a non-convex function, then the situation in the following figure could arise.*




### Proof of Strong Duality

To simplify, we assume $\mathcal{D}$ has nonempty interior, i.e., reint$\mathcal{D}$ = int$\mathcal{D}$ and $A$ is full rank.

When $f_0, ..., f_l$ are all convex, it is easy to show that the set

$$\mathcal{A} = \{(u, v, t) \in \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R} | \exists x \in \mathcal{D}, f_i(x) \leq u_i, i = 1, ..., l, v = Ax - b, f_0(x) \leq t\}$$

is convex. We define a second set $\mathcal{B}$ as

$$\mathcal{B} = \{(0, 0, s) \in \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R} | s < p^*\}.$$

We show that $\mathcal{A} \cap \mathcal{B} = \emptyset$. Indeed, suppose $(u, v, t) \in \mathcal{A} \cap \mathcal{B}$. By $(u, v, t) \in \mathcal{B}$, we know $u = 0, v = 0$ and $t < p^*$. By $(u, v, t) \in \mathcal{A}$, we know there exists $x \in \mathcal{D}$ such that $f_i(x) \leq 0, i = 1, ..., l, Ax = b$ and $f_0(x) \leq t < p^*$, which contradicts with the fact that $p^*$ is the minimum value for (11.3).

We can now invoke the separating hyperplane theorem: there exists $(\tilde{\lambda}, \tilde{\nu}, \mu) \neq 0$ and $\alpha$ such that

$$(u, v, t) \in \mathcal{A} \Longrightarrow (\tilde{\lambda}, \tilde{\nu}, \mu)'(u, v, t) \geq \alpha$$

and

$$(u, v, t) \in \mathcal{B} \Longrightarrow (\tilde{\lambda}, \tilde{\nu}, \mu)'(u, v, t) \leq \alpha$$

Since $u$ and $t$ are not bounded from above in $\mathcal{A}$, from the first implication we have $\tilde{\lambda} \geq 0$ and $\mu \geq 0$. From the second implication, we have $\mu t \leq \alpha$ for any $t < p^*$, and hence we have $\mu p^* \leq \alpha$. It follows that for any $x \in \mathcal{D}$

$$\sum_{i=1}^{l} \tilde{\lambda}_i f_i(x) + \tilde{\nu}'(Ax - b) + \mu f_0(x) \geq \alpha \geq \mu p^*.$$

If $\mu > 0$, then dividing both sides of the above inequality by $\mu$, we have

$$L(x, \tilde{\lambda}/\mu, \tilde{\nu}/\mu) = \sum_{i=1}^{l} \frac{\tilde{\lambda}_i}{\mu} f_i(x) + \frac{1}{\mu}\tilde{\nu}'(Ax - b) + f_0(x) \geq p^*.$$

Hence, $g(\tilde{\lambda}/\mu, \tilde{\nu}/\mu) = \inf_{x \in \mathcal{D}} L(x, \tilde{\lambda}/\mu, \tilde{\nu}/\mu) \geq p^*$. From weak duality, we know for any $(\lambda, \nu), \lambda \geq 0$, we have $g(\lambda, \nu) \leq p^*$. Therefore, we must have $g(\tilde{\lambda}/\mu, \tilde{\nu}/\mu) = p^*$—strong duality holds, and the dual optimal solution is $\lambda^* = \tilde{\lambda}/\mu, \nu^* = \tilde{\nu}/\mu$.

We next show that if Slater's condition is satisfied, then it is impossible for the normal vector of the separating hyperplane $(\tilde{\lambda}, \tilde{\nu}, \mu)$ to have $\mu = 0$. Suppose on the contrary that $\mu = 0$. We then have for any $x \in \mathcal{D}$

$$\sum_{i=1}^{l} \tilde{\lambda}_i f_i(x) + \tilde{\nu}'(Ax - b) \geq 0.$$

Let $\tilde{x} \in \text{int}\mathcal{D}$ be the point that satisfies the Slater's condition, i.e., $f_i(\tilde{x}) < 0, i = 1, ..., l$ and $A\tilde{x} = b$. We then have

$$\sum_{i=1}^{l} \tilde{\lambda}_i f_i(\tilde{x}) \geq 0,$$

which can only happen for $\tilde{\lambda} = 0$. As a result, we have for $\tilde{\nu} \neq 0$ and any $x \in \mathcal{D}$,

$$\tilde{\nu}'(Ax - b) \geq 0, \text{ and } \tilde{\nu}'(A\tilde{x} - b) = (\tilde{\nu}'A)\tilde{x} - \nu'b = 0.$$

Since $A$ is full rank and $\tilde{\nu} \neq 0$, it follows that $\tilde{\nu}'A \neq 0$. Let $\tilde{\nu}'A = (\tilde{a}_1, ..., \tilde{a}_n)$, then we have $\sum_{i=1}^{n} \tilde{a}_i \tilde{x}_i - \nu'b = 0$. Clearly, by $\tilde{x} \in \text{int}\mathcal{D}$, we can find $\epsilon \in \mathbb{R}^n$ such that $\tilde{x} + \epsilon \in \mathcal{D}$ and

$$\tilde{\nu}'(A(\tilde{x} + \epsilon) - b) = \sum_{i=1}^{n} \tilde{a}_i(\tilde{x}_i + \epsilon_i) - \nu'b < 0.$$

## Complementary Slackness and KKT Conditions

Suppose the primal and dual optimal values are attained and equal, and let $x^*$ and $(\lambda^*, \nu^*)$ be the primal and dual optimal solutions. We then have

$$\begin{aligned} f_0(x^*) &= g(\lambda^*, \nu^*) \\ &= \inf_{x \in \mathcal{D}} \left( f_0(x) + \sum_{i=1}^{l} \lambda_i^* f_i(x) + \sum_{i=1}^{m} \nu_i^* h_i(x) \right) \\ &\leq f_0(x^*) + \sum_{i=1}^{l} \lambda_i^* f_i(x^*) + \sum_{i=1}^{m} \nu_i^* h_i(x^*) \\ &\leq f_0(x^*). \end{aligned}$$

The above chain of inequalities implies the last two inequalities hold with equality. We can conclude from this that if $x^*$ and $(\lambda^*, \nu^*)$ are the primal and dual optimal solutions and strong duality holds, then

- the primal optimal solution $x^*$ also minimizes $L(x, \lambda^*, \nu^*)$;
- $\sum_{i=1}^{l} \lambda_i^* f_i(x^*) = 0 \Longrightarrow \lambda_i^* f_i(x^*) = 0, i = 1, ..., l.$

As in LP, the condition $\lambda_i^* f_i(x^*) = 0$ is referred to as the complementary slackness condition.

Conversely, if we have a pair of primal and dual feasible solutions $x^*$ and $(\lambda^*, \nu^*)$ such that

$$L(x^*, \lambda^*, \nu^*) = \inf_{x \in \mathcal{D}} L(x, \lambda^*, \nu^*), \ \lambda_i^* f_i(x^*) = 0, i = 1, ..., l,$$

then they must also be optimal with a duality gap zero. This is because

$$\begin{aligned}
&g(\lambda^*, \nu^*) \\
=&\inf_{x \in \mathcal{D}} L(x, \lambda^*, \nu^*) \\
=&f_0(x^*) + \sum_{i=1}^{l} \lambda_i^* f_i(x^*) + \sum_{i=1}^{m} \nu_i^* h_i(x^*) \\
=&f_0(x^*).
\end{aligned}$$

In particular when the problem is convex and differentiable, i.e., $f_i, i = 0, ..., l$ are convex and $h_i$ are affine, then the condition

$$L(x^*, \lambda^*, \nu^*) = \inf_{x \in \mathcal{D}} L(x, \lambda^*, \nu^*),$$

is equivalent as

$$\nabla f_0(x^*) + \sum_{i=1}^{l} \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^{m} \nu_i^* \nabla h_i(x^*) = \nabla f_0(x^*) + \sum_{i=1}^{l} \lambda_i^* \nabla f_i(x^*) + A'\nu^* = 0.$$

Therefore, when Slater's condition holds, the points $x^*$ and $(\lambda^*, \nu^*)$ are primal and dual optimal if and only if they satisfy

$$\begin{aligned}
&f_i(x^*) \le 0, i = 1, ..., l \\
&h_i(x^*) = 0, i = 1, ..., m, (Ax^* = b) \\
&\lambda^* \ge 0 \\
&\lambda_i^* f_i(x^*) = 0, i = 1, ..., l \\
&\nabla f_0(x^*) + \sum_{i=1}^{l} \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^{m} \nu_i^* \nabla h_i(x^*) = 0, (\nabla f_0(x^*) + \sum_{i=1}^{l} \lambda_i^* \nabla f_i(x^*) + A'\nu^* = 0).
\end{aligned}$$

The above set of conditions is referred to as *Karush-Kuhn-Tucker* (KKT) conditions.

**Example 11.4** *Consider the problem*

$$\begin{aligned}
\min \quad & \frac{1}{2} x' P x + q' x + r \\
s.t. \quad & Ax = b,
\end{aligned}$$

*where $P \in \mathbb{S}_+^n$. The KKT conditions for this problem are*

$$\begin{aligned}
&Ax^* = b \\
&Px^* + q + A'\nu^* = 0,
\end{aligned}$$

*which can be written as*

$$\begin{bmatrix} P & A' \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \nu^* \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix}.$$

*Solving this system of $m+n$ linear equations with $m+n$ variables gives the optimal primal and dual variables.*

## 11.3 Generalized Inequalities

We consider an optimization problem with generalized inequality constraints:

$$\begin{array}{rl} p^* = \min & f_0(x) \\ \text{s.t.} & f_i(x) \preceq_{K_i} 0, i = 1, ..., l \\ & h_i(x) = 0, i = 1, ..., m, \\ & x \in \mathbb{R}^n, \end{array} \tag{11.4}$$

where $K_i \subseteq \mathbb{R}^{k_i}$ are proper cones, and $f_i : \mathbb{R}^n \to \mathbb{R}^{k_i}$ are $K_i$-convex.

For generalized inequality $f_i(x) \preceq_{K_i} 0$, we associate a Lagrange multiplier *vector* $\lambda_i \in \mathbb{R}^{k_i}$ and define the Lagrangian as

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{l} \lambda_i' f_i(x) + \sum_{i=1}^{m} \nu_i h_i(x),$$

and the dual function

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu).$$

To qualify as a relaxation or valid lower bound, we require $\lambda_i' f_i(x) \leq 0$ for any $f_i(x)$ with $f_i(x) \preceq_{K_i} 0$. By definition of dual cone, this is equivalent as $\lambda_i \in K_i^*$ or $\lambda_i \succeq_{K_i^*} 0$. The Lagrange dual problem is

$$\begin{array}{rl} d^* = \max & g(\lambda, \nu) \\ \text{s.t.} & \lambda_i \succeq_{K_i^*} 0. \end{array} \tag{11.5}$$

It is easy to show that weak duality still holds: $d^* \leq p^*$.

**Example 11.5 (Semidefinite Program)** *Consider a semidefinite program of the form*

$$\begin{array}{rl} \min & c'x \\ s.t. & x_1 F_1 + ... + x_n F_n + G \preceq 0, \end{array}$$

*where $F_1, ..., F_n, G \in \mathbb{S}^k$. Note that here we have one generalized inequality: $f_1(x) \preceq_{K_1} 0$ with $f_1$ being a linear function and $K_1 = \mathbb{S}^k_+$. We associate this constraint with a dual variable $Z \in \mathbb{S}^k$, so the Lagrangian is*

$$L(x, Z) = c'x + tr((x_1 F_1 + ... + x_n F_n + G)Z) = (c_1 + tr(F_1 Z))x_1 + ... + (c_n + tr(F_n Z))x_n + tr(GZ).$$

*The dual function is then*

$$g(Z) = \begin{cases} tr(GZ), & \text{if} \quad c_i + tr(F_i Z) = 0, i = 1, ..., n, \\ -\infty, & \text{otherwise.} \end{cases}$$

*The dual problem is*

$$\begin{array}{rl} \max & tr(GZ) \\ \text{s.t.} & c_i + tr(F_i Z) = 0, i = 1, ..., n \\ & Z \succeq 0. \end{array}$$

**Example 11.6 (Standard Form Cone Program)** *Consider a cone program in standard form*

$$\begin{array}{rl} \min & c'x \\ s.t. & Ax = b, \\ & x \succeq_K 0. \end{array}$$

*The Lagrangian is*

$$L(x, \lambda, \nu) = (c' - \lambda' + \nu' A)x - \nu' b.$$

*The dual function is then*

$$g(\lambda, \nu) = \begin{cases} -\nu' b, & \textit{if} \quad c' - \lambda' + \nu' A = 0, i = 1, ..., n, \\ -\infty, & \textit{otherwise.} \end{cases}$$

*The dual problem is*

$$\begin{array}{rl} \max & -\nu' b \\ \text{s.t.} & -\lambda' + \nu' A = -c', \\ & \lambda \succeq_{K^*} 0. \end{array}$$

*By letting* $p' = -\nu'$ *and eliminating the slack variable* $\lambda$*, we then have the equivalent dual problem*

$$\begin{array}{rl} \max & p' b \\ \text{s.t.} & p' A \preceq_{K^*} c', \end{array}$$

*which is in the same form as the dual problem we derived for the standard form LP.*

Strong duality for problem (11.4) holds when $h_i(x), i = 1, ..., m$ are affine, $f_i(x)$ is $K_i$-convex, $f_0(x)$ is convex and a generalized version of Slater's condition holds: there exists an $x \in \text{relint}\mathcal{D}$ such that $Ax = b$ and $f_i(x) \prec_{K_i} 0$.

# References

[BV03] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2003.