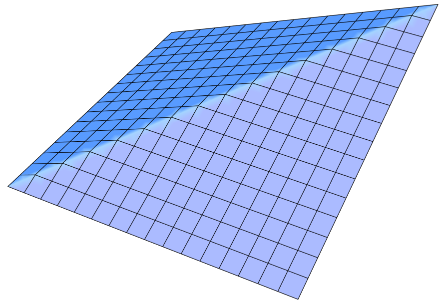


CO 250: Introduction to Optimization

Prof. Bertrand Guenin • Winter 2014 • University of Waterloo
TR 10:00AM–11:20AM in QNC 2502

Transcribed, LaTeXed, and edited by Rob Zimmerman

Note: These are personal lecture notes, not official course materials. They may contain errors (which by default you should attribute to me, Rob Zimmerman, rather than the course instructor). The permanent home of these — and all of my other lecture notes — is `https://rob-zimmerman.github.io/coursenotes`.

Contents

Lecture 1 — Tuesday, January 07, 2014

1. Formulations

We begin by framing optimization as a modelling discipline and by identifying the common structure shared by many applications.

Optimization Problems

Broadly speaking, **optimization** is the problem of minimizing or maximizing a quantity subject to constraints. These kinds of problems appear in many settings, such as planning GPS routes and parcel deliveries, choosing locations for hospitals or cell towers, and making pricing or investment decisions.

The basic theme of the course is that there are really two separate tasks involved in optimization: formulating the real-world problem mathematically, and then solving that mathematical formulation using a suitable algorithm.

The rough picture is therefore the following:

$$\boxed{\text{real-life optimization problem}} \longrightarrow \boxed{\text{mathematical formulation}}$$
$$\longrightarrow \boxed{\text{algorithm}} \longrightarrow \boxed{\text{optimal decision}}.$$

Suppose a company can produce three products using three machines. The relevant data are:

Product	Machine 1	Machine 2	Machine 3	Unit Sales Price
1	11	4	8	\$300
2	7	6	5	\$260
3	6	5	5	\$220
Resources	500	600	200	

We want to decide how many units of each product to produce in order to maximize total sales revenue, while respecting the available resources.

Let x_j be the number of units of product j produced, for $j = 1, 2, 3$.

Our **objective function** is

$$\max 300x_1 + 260x_2 + 220x_3.$$

Each machine column of the table gives one **constraint**:

$$\begin{aligned}
11x_1 + 7x_2 + 6x_3 &\le 500 && \text{(Machine 1)}\\
4x_1 + 6x_2 + 5x_3 &\le 600 && \text{(Machine 2)}\\
8x_1 + 5x_2 + 5x_3 &\le 200 && \text{(Machine 3)}\\
x_1, x_2, x_3 &\ge 0.
\end{aligned}$$

The full formulation is therefore

$$\begin{aligned}
\max \quad & 300x_1 + 260x_2 + 220x_3\\
\text{subject to} \quad & 11x_1 + 7x_2 + 6x_3 \le 500\\
& 4x_1 + 6x_2 + 5x_3 \le 600\\
& 8x_1 + 5x_2 + 5x_3 \le 200\\
& x_1, x_2, x_3 \ge 0.
\end{aligned}$$

A good formulation should match the word description exactly: every feasible production plan should give a feasible vector $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$, every feasible vector $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ should correspond to an allowed production plan, and the objective should measure exactly the quantity we want to optimize.

We can also write the formulation in matrix form:

$$\begin{aligned}
\max \quad & \begin{bmatrix} 300 \\ 260 \\ 220 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\
\text{subject to} \quad & \begin{bmatrix} 11 & 7 & 6 \\ 4 & 6 & 5 \\ 8 & 5 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \leq \begin{bmatrix} 500 \\ 600 \\ 200 \end{bmatrix}, \\
& \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.
\end{aligned}$$

(Here and in what follows, an inequality between two vectors of the same size means that the inequality holds componentwise.) This is our first example of a linear program. Next time we will abstract this example into a general production model and define linear programs formally.

Lecture 2 — Thursday, January 09, 2014

The numerical model above generalizes naturally to an indexed production planning problem.

Index products by $j \in \{1, \ldots, n\}$ and resources by $i \in \{1, \ldots, m\}$. Let c_j be the unit sales price of product j, let b_i be the amount of resource i available, and let A_{ij} be the amount of resource i required to produce one unit of product j. If x_j is the number of units of product j produced, then the general production model is

$$\begin{aligned}
\max \quad & \sum_{j=1}^{n} c_j x_j \\
\text{subject to} \quad & \sum_{j=1}^{n} A_{ij} x_j \leq b_i \quad (i = 1, \ldots, m), \\
& x_j \geq 0 \quad (j = 1, \ldots, n).
\end{aligned}$$

In matrix notation, let

$$\vec{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}, \qquad \vec{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}, \qquad \text{and} \qquad \vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix},$$

and let $A = [A_{ij}]$. Then the formulation becomes

$$\begin{aligned} \max \quad & \vec{c}^\top \vec{x} \\ \text{subject to} \quad & A\vec{x} \le \vec{b}, \\ & \vec{x} \ge \vec{0}. \end{aligned}$$

This is a simple example of a linear program.

Linear Programs

Before solving linear programs, we need a precise vocabulary for linear and affine expressions.

Definition 2.1 A **linear function** $f : \mathbb{R}^n \to \mathbb{R}$ has the form

$$f(\vec{x}) = \vec{a}^\top \vec{x} = a_1x_1 + \cdots + a_nx_n.$$

Example 2.2 For example, $f(\vec{x}) = x_1 - 3x_2$ is linear, whereas $f(\vec{x}) = x_1 - x_1x_2$ and $f(\vec{x}) = x_1 + x_2 + 4$ are not.

Definition 2.3 An **affine function** $f : \mathbb{R}^n \to \mathbb{R}$ has the form $\vec{a}^\top \vec{x} + \beta$ with $\beta \in \mathbb{R}$.

Obviously all linear functions are affine, but the converse does not hold.

Definition 2.4 A **linear constraint** is a relation of the form $f(\vec{x}) \le \beta$, $f(\vec{x}) \ge \beta$, or $f(\vec{x}) = \beta$, where f is linear.

Example 2.5 For example, $2x_1 - x_2 \le 3$ is linear, whereas $3x_1 - 2x_1x_2 = 4$ and $x_1 - x_2 < 3$ are not.

Standard linear programs use non-strict relations. A strict inequality can make the feasible region nonclosed, so even a bounded objective need not attain an optimum.

Definition 2.6 A **linear program** (**LP**) is an optimization problem that maximizes or minimizes an affine function subject to finitely many linear constraints.

Example 2.7 The following are linear programs:

1.
$$\begin{aligned} \max \quad & \vec{c}^\top \vec{x} \\ \text{subject to} \quad & A\vec{x} = \vec{b}, \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

2.
$$\begin{aligned} \max \quad & 2x_1 \\ \text{subject to} \quad & x_1 - x_2 = 0, \\ & x_1 + x_2 \geq 3, \\ & x_1 \geq 0. \end{aligned}$$

3.
$$\begin{aligned} \min \quad & 0 \\ \text{subject to} \quad & x_1 \geq 1, \\ & x_1 \leq 0. \end{aligned}$$

This last problem is stupid (or more precisely, infeasible), but it is still linear.

By contrast,
$$\begin{aligned} \min \quad & x_1 - x_2 + x_3 \\ \text{subject to} \quad & 2x_1 - x_2 = 4, \\ & x_1 + mx_2 \geq 7 \quad \text{for every } m \in [0, 6], \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$
is *not* a linear program, because it has infinitely many constraints (uncountably many, in fact). Standard linear programs have only finitely many constraints.

Linear programs are *nice*. Many interesting problems can be formulated as LPs, and large LPs with millions of variables or constraints can often be solved efficiently. Let us explore several real-world examples.

Example 2.8 (Pipeline procurement model)

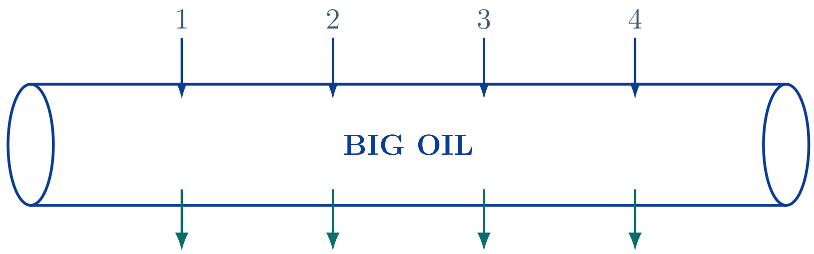


FIGURE 1

Outputs

Output	1	2	3	4
Demand	3500	3000	4000	5000

Inputs

Input	1	2	3	4
Capacity	4000	2000	3000	4500
Cost/barrel	70	50	90	70

The pipeline stages, together with the corresponding demands and capacities, are summarized in Figure 1.

The goal is to minimize the cost of buying oil while meeting all demands and respecting capacities. The pipeline initially contains 6000 barrels. Let p_i be the amount purchased from input i for $i = 1, \ldots, 4$, and let t_i be the amount in the pipeline right after output i. The resulting program is

$$\begin{aligned}
\min \quad & 70p_1 + 50p_2 + 90p_3 + 70p_4 \\
\text{subject to} \quad & p_1 \leq 4000, \quad p_2 \leq 2000, \quad p_3 \leq 3000, \quad \text{and} \quad p_4 \leq 4500, \\
& t_1 = 6000 + p_1 - 3500, \\
& t_2 = t_1 + p_2 - 3000, \\
& t_3 = t_2 + p_3 - 4000, \\
& t_4 = t_3 + p_4 - 5000, \\
& t_1, \ldots, t_4 \geq 0, \\
& p_1, \ldots, p_4 \geq 0.
\end{aligned}$$

Example 2.9 (Fire station locations) A city needs to build fire stations so that every school is within 2 km of a fire station. The building costs are

Location	L_1	L_2	L_3	L_4
Cost (in millions)	3	5	2	4

and the school-to-location distance table is

	S_1	S_2	S_3	S_4
L_1	5	1.5	1.3	2.1
L_2	0.9	5	0.1	4
L_3	3	0.6	1.1	0.4
L_4	0.3	0.6	2.8	0.6

Each entry gives the distance from a candidate fire station location to a school.

Lecture 3 — Tuesday, January 14, 2014

We continue on from Example 2.9.

Example 3.1 (Fire station locations (continued)) The goal is to choose fire station locations so that the total building cost is minimized and every school is within 2 km of a fire station, given the data in Example 2.9. Let

$$x_j = \begin{cases} 1, & \text{if a station is built at location } j, \\ 0, & \text{otherwise.} \end{cases}$$

Then the formulation is

$$\begin{aligned} \min \quad & 3x_1 + 5x_2 + 2x_3 + 4x_4 \\ \text{subject to} \quad & x_j \geq 0, \qquad x_j \leq 1, \qquad \text{and} \qquad x_j \in \mathbb{Z}, \quad (j = 1, 2, 3, 4), \\ & x_2 + x_4 \geq 1 \quad \text{(school 1 within 2 km of a station)}, \\ & x_1 + x_3 + x_4 \geq 1 \quad \text{(school 2 within 2 km of a station)}, \\ & x_1 + x_2 + x_3 \geq 1 \quad \text{(school 3 within 2 km of a station)}, \\ & x_3 + x_4 \geq 1 \quad \text{(school 4 within 2 km of a station)}. \end{aligned}$$

To check that this formulation is correct, we must verify both directions of the translation between words and mathematics.

$$\boxed{\text{Mathematical formulation}} \longleftrightarrow \boxed{\text{word description}}$$

Any vector $\vec{x}$ satisfying all constraints corresponds to a valid decision about whether to build, and every valid decision of this kind gives such a vector. Both directions are needed when we check formulation correctness.

The optimization problem in Example 3.1 is not a linear program because of the integrality constraint $x_j \in \mathbb{Z}$. It is an **integer program** (**IP**), obtained by taking an LP and imposing integrality on a nonempty set of variables. Integer programs are very general and can be difficult to solve, although many important practical problems are modelled this way.

Example 3.2 (Binary assignment problem) We begin with a small qualification table to motivate the binary assignment variables and constraints.

	J_1	J_2	J_3
S_1	✓	X	X
S_2	X	✓	✓
S_3	✓	X	✓

X = not qualified, ✓ = qualified.

The goal is to assign as many students as possible to jobs for which they are qualified, while ensuring that each student gets at most one job and each job is assigned to at most one student.

Let $x_{ij} = 1$ if student i is assigned to job j, and $x_{ij} = 0$ otherwise.

$$\begin{aligned} \max \quad & \sum_{i=1}^{3} \sum_{j=1}^{3} x_{ij} \\ \text{subject to} \quad & x_{ij} \in \{0,1\} \quad \text{for } i,j = 1,2,3 \\ & x_{i1} + x_{i2} + x_{i3} \le 1 \quad \text{for } i = 1,2,3 \\ & x_{1j} + x_{2j} + x_{3j} \le 1 \quad \text{for } j = 1,2,3 \\ & x_{12} = 0, \qquad x_{21} = 0, \qquad x_{13} = 0, \qquad \text{and} \qquad x_{32} = 0 \end{aligned}$$

Example 3.3 (Transportation model) This model captures how supply, demand, and shipping costs interact in a network of warehouses and stores.

Let warehouses be indexed by $i = 1, 2, \ldots, m$ and stores by $j = 1, 2, \ldots, n$. Let c_{ij} be the cost of transporting one unit from warehouse i to store j, let u_i be the number of units available at warehouse i, and let b_j be the number of units requested at store j.

The goal is to minimize transportation costs while respecting warehouse capacities and meeting store demands.

Let x_{ij} be the number of units going from warehouse i to store j. Then the problem is

$$\begin{aligned} \text{min} \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \\ \text{subject to} \quad & \sum_{j=1}^{n} x_{ij} \leq u_i \quad \text{for } i = 1, \ldots, m \\ & \sum_{i=1}^{m} x_{ij} = b_j \quad \text{for } j = 1, \ldots, n \\ & x_{ij} \geq 0 \quad \text{for } j = 1, \ldots, n, \; i = 1, \ldots, m \end{aligned}$$

We can make the problem more realistic by introducing a fixed cost α_i for opening warehouse i. If warehouse i is open, it has capacity u_i; otherwise, its capacity is 0. Let $y_i = 1$ if warehouse i is open, and let $y_i = 0$ otherwise. The modified model is

$$\begin{aligned} \text{min} \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} \alpha_i y_i \\ \text{subject to} \quad & \sum_{j=1}^{n} x_{ij} \leq u_i y_i \quad \text{for } i = 1, \ldots, m \\ & \sum_{i=1}^{m} x_{ij} = b_j \quad \text{for } j = 1, \ldots, n \end{aligned}$$

$$x_{ij} \geq 0 \quad \text{for } i = 1, \ldots, m,\ j = 1, \ldots, n$$
$$y_i \in \{0, 1\} \quad \text{for } i = 1, \ldots, m.$$

Lecture 4 — Thursday, January 16, 2014

Modelling Tricks

We now collect practical modelling patterns that repeatedly appear in linear and integer optimization. Before looking at specific tricks, it helps to remember that each reformulation should preserve feasibility and objective meaning.

Minimizing a maximum. Many scheduling and balancing problems reduce to controlling the largest of several quantities.

Suppose we want to solve
$$\min_{\vec{x} \in \mathbb{R}^n} \max\{f_1(\vec{x}), \ldots, f_k(\vec{x})\},$$
and let l^* denote the optimal value.

For example, if $f_i(\vec{x})$ is the completion time for task i, then $\max\{f_1(\vec{x}), \ldots, f_k(\vec{x})\}$ is the finishing time of the last task, and l^* is the earliest time at which all tasks are complete. Finding l^* is equivalent to minimizing y subject to $y \geq f_1(\vec{x}), \ldots, y \geq f_k(\vec{x})$, since at optimality we have $y = \max\{f_1(\vec{x}), \ldots, f_k(\vec{x})\}$, so

$$\min_{\vec{x} \in \mathbb{R}^n} \max\{f_1(\vec{x}), \ldots, f_k(\vec{x})\} \iff \begin{array}{rl} \min & y \\ \text{subject to} & y \geq f_1(\vec{x}) \\ & \vdots \\ & y \geq f_k(\vec{x}) \end{array}$$

Modelling absolute values. Absolute values can often be modelled by reusing the same reformulation based on a maximum. If $x \in \mathbb{R}$ is a variable and we want to minimize $|x|$, we can write

$|x| = \max\{x, -x\}$ and apply the same trick, so

$$\min_{x\in\mathbb{R}} |x| \iff \begin{array}{rl} \min & y \\ \text{subject to} & y \geq x \\ & y \geq -x \end{array}$$

Modelling OR constraints. Logical conditions such as OR statements can also be translated into integer constraints.

Suppose we want to minimize some linear function $f(\vec{x})$ subject to $x_1, x_2, x_3 \geq 0$, together with

1. $2x_1 + 3x_2 + x_3 \geq 5$, or
2. $4x_1 + 2x_2 + 8x_3 \geq 8$.

We want *at least* one of (1) or (2) to hold. Let $y \in \{0, 1\}$ select which condition is enforced, where

$$y = \begin{cases} 1, & \text{if we enforce (1)}, \\ 0, & \text{if we enforce (2)}. \end{cases}$$

We then modify the constraints as follows:

1'. $2x_1 + 3x_2 + x_3 \geq 5y$. This trick works because $x_1, x_2, x_3 \geq 0$.

2'. $4x_1 + 2x_2 + 8x_3 \geq 8(1 - y)$.

So the equivalent IP is

$$\begin{array}{rl} \min & f(\vec{x}) \\ \text{subject to} & 2x_1 + 3x_2 + x_3 \geq 5y \\ & 4x_1 + 2x_2 + 8x_3 \geq 8(1 - y) \\ & y \in \{0, 1\} \end{array}.$$

For this particular formulation, nonnegative coefficients and variables ensure that the unselected constraint becomes automatic. More generally, the same construction works whenever the left-hand side of an unselected constraint is known to be nonnegative; otherwise one uses an appropriate big-M bound.

A similar idea works with three constraints. Suppose we want to minimize some linear function $f(\vec{x})$ subject to $x_1, x_2, x_3 \geq 0$, together with

1. $x_1 + 2x_2 + x_3 \geq 2$, or
2. $2x_1 + 4x_2 \geq 6$, or
3. $2x_2 + 4x_3 \geq 8$.

We want *at least two* of (1), (2), and (3) to hold. For $i = 1, 2, 3$, let $y_i \in \{0, 1\}$, where $y_i = 1$ means that we select constraint i to be enforced. Then we require $y_1 + y_2 + y_3 \geq 2$ and modify the constraints as follows:

1'. $x_1 + 2x_2 + x_3 \geq 2y_1$

2'. $2x_1 + 4x_2 \geq 6y_2$

3'. $2x_2 + 4x_3 \geq 8y_3$

So the equivalent IP is

$$\begin{array}{rl} \min & f(\vec{x}) \\ \text{subject to} & x_1 + 2x_2 + x_3 \geq 2y_1 \\ & 2x_1 + 4x_2 \geq 6y_2 \\ & 2x_2 + 4x_3 \geq 8y_3 \\ & y_1 + y_2 + y_3 \geq 2 \\ & y_1, y_2, y_3 \in \{0, 1\} \end{array}.$$

This idea generalizes to k constraints.

Discrete-valued variables. When a variable is restricted to a finite set of values, binary selectors provide a clean encoding.

Suppose we want $x \in \{3, 7, 29, 86\}$. Write $x = 3y_1 + 7y_2 + 29y_3 + 86y_4$, with $y_1, \ldots, y_4 \in \{0, 1\}$ and

$$\sum_{i=1}^{4} y_i = 1.$$

This simple trick is surprisingly powerful.

We next apply the formulation viewpoint to shortest paths and models based on cuts.

The Shortest Path Problem

To motivate the formulation, we first fix the concrete network in Figure 2, which we will use throughout this discussion.

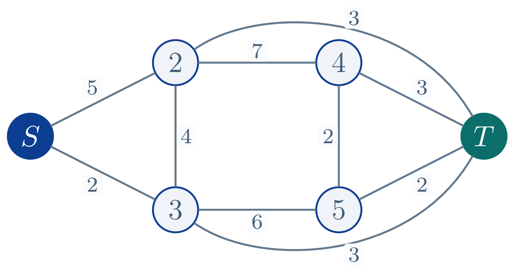


FIGURE 2

Finding the shortest route from S to T is the problem that a GPS solves when it is asked for a route between two places. We now develop the underlying graph-theoretic language behind it.

> **Definition 4.1** A **graph** G is a pair (V, E), where V is a finite set of objects called **vertices** and E is a set of unordered pairs of vertices.

Thus the network in Figure 2 is a graph: E is the set of roads, and V is the set of meeting points. For the smaller example in Figure 3, let $V = \{1, 2, 3, 4, 5\}$ and $E = \{12, 32, 14, 54, 42\}$.

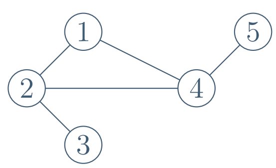


FIGURE 3

> **Definition 4.2** Let $G = (V, E)$ be a graph. A **path** P in G is a sequence of edges of the form $u_1u_2, u_2u_3, \ldots, u_{k-1}u_k \in E$. Additionally, we impose that $u_1, \ldots, u_k$ are all distinct. We say that P is a $\boldsymbol{u_1u_k}$**-path**.

It is often useful to think of a graph as a set of edges.

The **shortest path problem** is as follows. Let $G = (V, E)$ be a graph, let $s, t \in V$ with $s \neq t$ (think of these vertices as the $s$tart and end vertices), and let $c_e \geq 0$ for all $e \in E$ (think of c_e as the $c$ost of travelling along edge e). Let P be a path in G and define

$$c(P) := \sum_{e \in P} c_e,$$

which is the **total edge length** (i.e., the total cost of traveling). We want to minimize $c(P)$ subject to P being an st-path.

Our goal is to formulate this problem as an IP. There are two pieces of bad news. First, the process is going to get messy. Second, the resulting IP will look odd. The good news is that we will develop theory to handle this.

Cuts and st-Cuts

Cuts are the structural bridge between paths and linear constraints, so we introduce them explicitly before formulating the model.

Definition 4.3 Let $G = (V, E)$ be a graph. Let $U \subsetneq V$, where $U \neq \emptyset$. Let

$$\delta(U) := \{uv \in E \mid u \in U, v \notin U\}.$$

We call $\delta(U)$ a **cut**.

Example 4.4

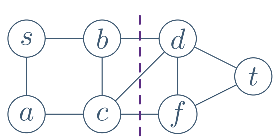


FIGURE 4

In Figure 4, if $U = \{s, a, c, b\}$ then $\delta(U) = \{bd, cd, cf\}$.

Example 4.5 Consider the graph and highlighted cut in Figure 5.

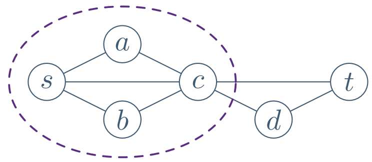


FIGURE 5

Let $U = \{s, a, b, c\}$. Then $\delta(U) = \{cd, ct\}$.

Removing the cut edges leaves the graph in Figure 6.

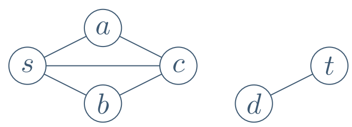


FIGURE 6

Definition 4.6 A cut $\delta(U)$ is an ***st*-cut** if it is a cut and $s \in U, t \notin U$.

Every st-path and every st-cut must share at least one edge.

Lecture 5 — Tuesday, January 21, 2014

We now connect graph cuts to path structure and use that connection to build optimization formulations.

We start with the picture in Figure 7 to keep the path–cut interaction concrete.

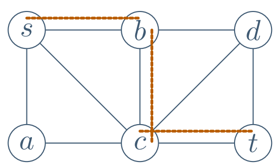


FIGURE 7

Our st-path here is $P = \{sb, bc, ct\}$, and our st-cut is $\delta(\{s, a, c, b\}) = \{bd, cd, ct\}$. This cut consists of the edges with one endpoint in $\{s, a, c, b\}$ and the other endpoint outside that set.

Lemma 5.1 Let P be an st-path and let $\delta(U)$ be an st-cut. Then $P \cap \delta(U) \neq \emptyset$.

The crossing edge used in the proof is isolated in Figure 8.

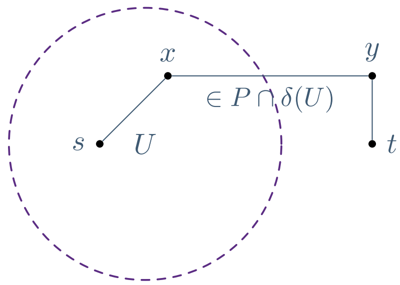


FIGURE 8

Proof. Follow the path P from s, and let x be the last vertex of P in U. Such a vertex exists because $s \in U$. Since $t \notin U$, the path has a next vertex y, and $y \notin U$. Thus $xy \in \delta(U) \cap P$. $\square$

An st-path P contains at least one edge from every st-cut. A natural question is whether every set of edges that contains at least one edge from every st-cut must itself be an st-path.

In general, the answer is no, as Figure 9 shows.

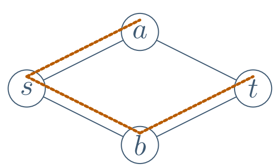


FIGURE 9

If we take one edge from every st-cut, the chosen set need not itself be an st-path.

Theorem 5.2 Let B be a set of edges that contains one edge from every st-cut. Then B contains an st-path.

Proof. For a contradiction, suppose there is no st-path $P \subseteq B$. Let

$$U = \{s\} \cup \{u \mid \text{there exists an } su\text{-path } Q \subseteq B\}.$$

Then $t \notin U$. By hypothesis, B contains an edge xy of the st-cut $\delta(U)$; we may assume $x \in U$ and $y \notin U$. Since $x \in U$, there is an sx-path $Q \subseteq B$. Appending xy to Q gives an sy-path $Q' \subseteq B$, so $y \in U$—a contradiction. $\square$

With the cut machinery in place, we can now write an optimization model for shortest paths.

Let $G = (V, E)$ be a graph. Let $s, t \in V$ with $s \neq t$. Let $c_e \geq 0$ for all $e \in E$. The goal is to find an st-path P minimizing

$$c(P) = \sum_{e \in P} c_e.$$

We choose $B \subseteq E$ so that the following two conditions hold:

(1) B contains at least one edge from every st-cut.

(2) B minimizes

$$c(B) = \sum_{e \in B} c_e.$$

These conditions force B to contain a shortest st-path. Indeed, the theorem above gives an st-path $P \subseteq B$, so nonnegativity of the edge costs gives $c(P) \leq c(B)$. Conversely, every st-path is itself a feasible choice of B. Hence an optimal B cannot cost more than a shortest path, and the two inequalities force equality; the path $P \subseteq B$ is therefore shortest.

Accordingly, let

$$x_e = \begin{cases} 1 & \text{if } e \in B, \\ 0 & \text{otherwise.} \end{cases}$$

We then want to minimize

$$\sum_{e \in E} c_e x_e.$$

The key constraint is that $B \cap \delta(U)$ contains at least one edge, which we write as

$$\sum_{e \in \delta(U)} x_e \geq 1.$$

We also impose $x_e \in \{0,1\}$. Formally, the shortest path problem is formulated as the following IP:

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e x_e \\ \text{subject to} \quad & \sum_{e \in \delta(U)} x_e \geq 1 \quad \text{for all } U \subseteq V, s \in U, t \notin U \\ & x_e \in \{0,1\} \qquad \text{for all } e \in E. \end{aligned}$$

Example 5.3

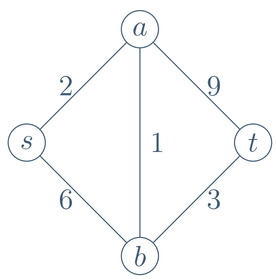


Figure 10

For the weighted graph in Figure 10, our objective function is

$$\min 2x_{sa} + 6x_{sb} + 3x_{bt} + 9x_{at} + 1x_{ab}.$$

$$\begin{aligned} \text{subject to } (U = \{s\}) \quad & x_{sa} + x_{sb} \geq 1 \\ (U = \{s,a\}) \quad & x_{sb} + x_{at} + x_{ab} \geq 1 \\ (U = \{s,b\}) \quad & x_{sa} + x_{ab} + x_{bt} \geq 1 \\ (U = \{s,a,b\}) \quad & x_{at} + x_{bt} \geq 1 \end{aligned}$$

If we were to solve this, we would get the solution

$$\bar{x}_{sa} = \bar{x}_{ab} = \bar{x}_{bt} = 1$$

and $\bar{x}_{at} = \bar{x}_{sb} = 0$. Therefore $B = \{sa, ab, bt\}$ is an st-path, and in this example it is the shortest one.

There is a slight problem with this idea: if there are 100 vertices, then there are 2^{98} such cut constraints. This is too many to write explicitly. Fortunately, we will see that this does not actually cause a problem.

Lecture 6 — Thursday, January 23, 2014

We now broaden the perspective from linear models to nonlinear ones and clarify the core notions needed to analyze them.

Nonlinear Programs

We begin with the abstract template that captures a broad class of constrained nonlinear models.

Definition 6.1 Let

$$f : \mathbb{R}^n \to \mathbb{R} \qquad \text{and} \qquad g_i : \mathbb{R}^n \to \mathbb{R}, \quad i = 1, \ldots, m.$$

A **nonlinear program** has the form

$$\begin{aligned} \min \quad & f(\vec{x}) \\ \text{subject to} \quad & g_i(\vec{x}) \leq 0 \quad (i = 1, \ldots, m). \end{aligned}$$

Nonlinear programs arise naturally when penalties or response functions are nonlinear.

Example 6.2 (Safe operating conditions) Suppose the system $A\vec{x} \leq \vec{b}$ describes safe operating conditions for a nuclear reactor. The slack in constraint i is

$$b_i - \text{row}_i(A)\vec{x} \geq 0 \quad (i \in \{1, \ldots, m\}).$$

If there exists i such that $b_i - \text{row}_i(A)\vec{x} < 0$, then the corresponding safety constraint is violated, the reactor goes into meltdown, and the city is destroyed.

To stay far from meltdown conditions, we want a solution that remains well inside the feasible region (i.e., the solution should be “robust”). One way to model this is to use a penalty function that grows rapidly as a slack approaches 0 (for example $f(\alpha) = \frac{1}{\alpha}$ or $f(\alpha) = -\log(\alpha)$).

Example 6.3 (Pricing) We want to choose a price that maximizes revenue.

Suppose we set a price p with $50 \le p \le 70$ for a fancy new e-reader.

Price	50	60	70
demand (in 1000s)	90	80	20

We fit a demand function through these points, such as $d(p) = ap^2 + bp + c$. An Excel regression gives

$$d(p) = -0.25p^2 + 26.5p - 610.$$

The resulting model is

$$\begin{aligned} \max \quad & p \cdot d(p) = -0.25p^3 + 26.5p^2 - 610p \\ \text{subject to} \quad & 50 \le p \le 70. \end{aligned}$$

Feasibility, Optimality, and Unboundedness

At this point, the notion of “solving” still needs to be clarified.

Example 6.4

$$\begin{aligned} \max \quad & 2x_1 - x_2 \\ \text{subject to} \quad & x_1 + x_2 = 1, \\ & x_1, x_2 \ge 0. \end{aligned}$$

Here the optimum is clearly $x_1 = 1$, $x_2 = 0$ and there is no ambiguity.

Example 6.5 Consider

$$\begin{aligned} \min \quad & x^2 \\ \text{subject to} \quad & x \geq 1, \\ & x \leq 0. \end{aligned}$$

Here what it means to "solve" is not so clear because the problem is infeasible.

Definition 6.6 An assignment of values to the variables is a **feasible solution** if all constraints are satisfied.

Definition 6.7 An optimization problem is **infeasible** if it has no feasible solution. Otherwise, it is **feasible**.

Definition 6.8 For a maximization problem, an **optimum solution** is a feasible solution for which no other feasible solution has a larger objective value. The minimization case is analogous.

Definition 6.9 A maximization problem is **unbounded** if for every $M \in \mathbb{R}$, there exists a feasible solution with objective value at least M. For a minimization problem, unboundedness means that for every $M \in \mathbb{R}$, some feasible solution has objective value at most M.

A feasible optimization problem may fail to attain an optimum without being unbounded:

Example 6.10 Consider

$$\begin{aligned} \min \quad & \frac{1}{x} \\ \text{subject to} \quad & x \geq 1 \end{aligned}.$$

This problem has no optimum solution: if $x_1 = m$ were optimal, then $x_1' = m + 1$ would give $\frac{1}{(m+1)} < \frac{1}{m}$, a contradiction. Neither is this problem infeasible, nor is it unbounded.

For LPs, however, these three properties do form a trichotomy:

Theorem 6.11 (Fundamental theorem of linear programming) For any linear program, exactly one of the following holds:

1. The problem has an optimal solution.
2. The problem is infeasible.
3. The problem is unbounded.

The fundamental theorem is proved as a consequence of the **two-phase simplex method**, which we will examine soon.

Figure 11 illustrates all three cases.

FIGURE 11

2. Solving Linear Programs: the Simplex Method

The simplex method does not directly involve simplices.

The next lectures develop the simplex algorithm through certificates, pivots, and canonical representations.

Certificates of Infeasibility

Before running simplex mechanically, we need a way to certify when no feasible point exists.

Example 6.12 Consider

$$\begin{aligned} \max \quad & \vec{c}^{\top}\vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 3 & -2 & -6 & 7 \\ 2 & -1 & -2 & 4 \end{bmatrix} \vec{x} = \begin{bmatrix} -6 \\ -4 \end{bmatrix} \quad \begin{matrix} (1) \\ (2) \end{matrix} \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

We prove that this system has no feasible solution.

Subtract equation ① from twice equation ② to get

$$\begin{bmatrix}1 & 0 & 2 & 1\end{bmatrix} \vec{x} = -2$$

This is a valid linear combination of the original equations, so it must also hold. However, the left-hand side is nonnegative when $\vec{x} \geq \vec{0}$, while the right-hand side is -2. That is impossible, so the system has no feasible solution.

In Example 6.12, we found a vector $\vec{y}$ such that $\vec{y}^\top A \geq 0$ and $\vec{y}^\top \vec{b} < 0$:

$$\underbrace{\begin{bmatrix}-1 \\ 2\end{bmatrix}^\top}_{\vec{y}} \underbrace{\left(\begin{bmatrix}3 & -2 & -6 & 7 \\ 2 & -1 & -2 & 4\end{bmatrix} \vec{x}\right)}_{A\vec{x}} = \underbrace{\begin{bmatrix}-1 \\ 2\end{bmatrix}^\top \begin{bmatrix}-6 \\ -4\end{bmatrix}}_{\vec{y}^\top \vec{b} < 0}$$

$$\vec{y}^\top A \geq 0 \qquad \text{and} \qquad \vec{y}^\top \vec{b} < 0$$

Thus $\vec{y}$ acts as a **certificate of infeasibility**.

Theorem 6.13 If there exists a vector $\vec{y}$ such that $\vec{y}^\top A \geq 0$ and $\vec{y}^\top \vec{b} < 0$, then the system $A\vec{x} = \vec{b}$, $\vec{x} \geq \vec{0}$ has no feasible solution.

Proof. Let $\bar{\vec{y}}$ satisfy the two certificate inequalities, and suppose for contradiction that a feasible vector $\bar{\vec{x}}$ exists. Then

$$\underbrace{\bar{\vec{y}}^\top A}_{\geq 0} \underbrace{\bar{\vec{x}}}_{\geq 0} = \underbrace{\bar{\vec{y}}^\top \vec{b}}_{< 0}$$

which is impossible. This contradiction shows that no feasible $\bar{\vec{x}}$ can exist. $\square$

Lecture 7 — Tuesday, January 28, 2014

Certificates of Unboundedness

A second failure mode is unboundedness, and it is certified by a feasible improving direction.

Example 7.1

$$\begin{aligned} \max \quad & z = \begin{bmatrix} 1 & 3 & 0 & 0 & 1 \end{bmatrix} \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 1 & 3 & -1 & 1 & 0 \\ -2 & 4 & 1 & 0 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

The idea is to construct a parameterized vector $\vec{x}(t)$ for every $t \geq 0$ such that

(A) $\vec{x}(t)$ is feasible for every $t \geq 0$, and

(B) $z \to \infty$ as $t \to \infty$.

A useful choice is the following:

$$\vec{x}(t) = \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 3 \end{bmatrix}}_{\bar{\vec{x}}} + t \underbrace{\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}}_{\vec{r}} = \bar{\vec{x}} + t\vec{r}.$$

For example, $\vec{x}(0) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 3 \end{bmatrix}$ and $\vec{x}(1) = \begin{bmatrix} 2 \\ 0 \\ 1 \\ 1 \\ 4 \end{bmatrix}$.

We now check the two required properties.

(A) $A\vec{x}(t) = A\bar{\vec{x}} + tA\vec{r} = \vec{b}$. Notice that $\bar{\vec{x}}, \vec{r} \geq \vec{0}$ and $t \geq 0$, so $\vec{x}(t) \geq \vec{0}$.

(B) $\vec{c}^\top \vec{x}(t) = \vec{c}^\top(\bar{\vec{x}} + t\vec{r}) = \vec{c}^\top \bar{\vec{x}} + t\vec{c}^\top \vec{r}$. Note that $\vec{c}^\top \bar{\vec{x}} \in \mathbb{R}$, and $\vec{c}^\top \vec{r} > 0$, so $t\vec{c}^\top \vec{r} \to \infty$.

In Example 7.1, the vector $\vec{x}(t)$ acts as a **certificate of unboundedness**.

Proposition 7.2 The LP

$$\begin{aligned} \max \quad & \vec{c}^\top \vec{x} \\ \text{subject to} \quad & A\vec{x} = \vec{b}, \\ & \vec{x} \geq \vec{0} \end{aligned}$$

is unbounded if there exist a feasible vector $\bar{\vec{x}}$ and a direction vector $\vec{r}$ such that the following conditions hold:

(1) $A\bar{\vec{x}} = \vec{b}$

(2) $A\vec{r} = 0$

(3) $\bar{\vec{x}} \geq \vec{0}$

(4) $\vec{r} \geq \vec{0}$

(5) $\vec{c}^\top \vec{r} > 0$

Proof. For each $t \geq 0$, define

$$\vec{x}(t) := \bar{\vec{x}} + t\vec{r}.$$

By assumptions (1) and (2),

$$A\vec{x}(t) = A\bar{\vec{x}} + tA\vec{r} = \vec{b} + t\vec{0} = \vec{b}.$$

By assumptions (3) and (4), both $\bar{\vec{x}}$ and $\vec{r}$ are componentwise nonnegative, so $\vec{x}(t) \geq \vec{0}$ for every $t \geq 0$. Hence $\vec{x}(t)$ is feasible for every $t \geq 0$. The objective value at $\vec{x}(t)$ is

$$\vec{c}^\top \vec{x}(t) = \vec{c}^\top \bar{\vec{x}} + t\vec{c}^\top \vec{r}.$$

Since $\vec{c}^\top \vec{r} > 0$ by assumption (5), this quantity tends to ∞ as $t \to \infty$. Therefore the LP has feasible solutions with arbitrarily large objective value, so it is unbounded. $\square$

Standard Equality Form

To develop the simplex algorithm systematically, we first put linear programs into a common canonical template.

Definition 7.3 The following LP is in **standard equality form** (**SEF**):

$$\begin{aligned} \max \quad & \vec{c}^{\top}\vec{x} \\ \text{subject to} \quad & A\vec{x} = \vec{b} \\ & \vec{x} \geq \vec{0} \end{aligned}$$

The simplex algorithm requires the LP to be in SEF, and fortunately this is always possible. We therefore solve an arbitrary LP, given a program (P), as follows.

Step 1: Convert (P) into (P'), an LP in SEF.

Step 2: Solve (P') with the simplex algorithm.

Step 3: Use the (P') result to recover a result for (P).

We thus need (P) and (P') to be "equivalent". What do we mean by that? We have the following desiderata:

1 (P) is infeasible if and only if (P') is infeasible.

2 (P) is unbounded if and only if (P') is unbounded.

3 We can use an optimum solution of (P') to construct an optimum solution for (P), and vice versa.

Proposition 7.4 Any linear program can be put into SEF.

Proof. Negate the objective of a minimization problem to obtain a maximization problem. Add a nonnegative slack variable to each $\leq$ constraint, and subtract a nonnegative surplus variable from each $\geq$ constraint, to turn inequalities into equalities. Finally, replace every free variable x_j by $x_j^+ - x_j^-$ with $x_j^+, x_j^- \geq 0$. Each operation preserves feasible solutions and objective values in both directions (after undoing the introduced variables), so the resulting SEF program is equivalent to the original one. $\square$

Example 7.5

$$\min \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix} \cdot \vec{x}$$

is equivalent to

$$\max \begin{bmatrix} -2 \\ 1 \\ -1 \end{bmatrix} \cdot \vec{x}.$$

Example 7.6

$$\begin{aligned} \max \quad & \ldots \\ \text{subject to} \quad & x_1 + x_2 - x_3 \le 5 \end{aligned}$$

is equivalent to

$$\begin{aligned} \max \quad & \ldots \\ \text{subject to} \quad & x_1 + x_2 - x_3 + x_4 = 5, \\ & x_4 \ge 0. \end{aligned}$$

Example 7.7 Consider

$$\begin{aligned} \max \quad & \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 1 & 2 & 1 \\ -1 & 1 & 3 \end{bmatrix} \vec{x} = \begin{bmatrix} 6 \\ 4 \end{bmatrix}, \\ & x_1, x_2 \ge 0. \end{aligned}$$

Here, x_3 is a **free variable**. That is, it is not specified whether it should be positive or negative. We replace it by $x_3^+ - x_3^-$, where $x_3^+, x_3^- \ge 0$. So:

$$\begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = -x_1 + 2x_2 + x_3 = -x_1 + 2x_2 + x_3^+ - x_3^- = \begin{bmatrix} -1 \\ 2 \\ 1 \\ -1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3^+ \\ x_3^- \end{bmatrix}$$

The reformulated LP is therefore

$$\begin{aligned} \max \quad & \begin{bmatrix} -1 \\ 2 \\ 1 \\ -1 \end{bmatrix} \cdot \vec{x}' \\ \text{subject to} \quad & \begin{bmatrix} 1 & 2 & 1 & -1 \\ -1 & 1 & 3 & -3 \end{bmatrix} \vec{x}' = \begin{bmatrix} 6 \\ 4 \end{bmatrix}, \\ & \vec{x}' = \begin{bmatrix} x_1 \\ x_2 \\ x_3^+ \\ x_3^- \end{bmatrix} \geq \vec{0}. \end{aligned}$$

Does our reformulation work?

$$\begin{aligned} \begin{bmatrix} 1 & 2 & 1 \\ -1 & 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} &= x_1 \begin{bmatrix} 1 \\ -1 \end{bmatrix} + x_2 \begin{bmatrix} 2 \\ 1 \end{bmatrix} + x_3 \begin{bmatrix} 1 \\ 3 \end{bmatrix} \\ &= x_1 \begin{bmatrix} 1 \\ -1 \end{bmatrix} + x_2 \begin{bmatrix} 2 \\ 1 \end{bmatrix} + x_3^+ \begin{bmatrix} 1 \\ 3 \end{bmatrix} - x_3^- \begin{bmatrix} 1 \\ 3 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 2 & 1 & -1 \\ -1 & 1 & 3 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3^+ \\ x_3^- \end{bmatrix}. \end{aligned}$$

Our reformulation works!

There are therefore three reasons an LP may fail to be in SEF: minimization instead of maximization, inequality constraints, and free variables. We now know how to handle all three.

Improving Feasible Solutions

The simplex algorithm can be viewed as a repeatable improvement loop starting from feasibility:

(1) Find a feasible solution.

(2) Try to get a better feasible solution.

(3) Repeat (2) until either our solution is optimal, or we notice the LP is unbounded.

As a first concrete iteration, we now carry out one explicit pivot to illustrate how objective improvement and feasibility are balanced.

Example 7.8

$$\begin{aligned}
\max \quad & z = \begin{bmatrix} 4 \\ 5 \\ 0 \\ 0 \end{bmatrix} \cdot \vec{x} \\
\text{subject to} \quad & \begin{bmatrix} 2 & 2 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 8 \\ 10 \end{bmatrix}, \\
& \vec{x} \geq \vec{0}.
\end{aligned}$$

By inspection, a feasible solution is $x = \begin{bmatrix} 0 \\ 0 \\ 8 \\ 10 \end{bmatrix}$, which gives $z = 0$.

Our goal is to find a feasible solution x' with $z > 0$. Set $x'_1 = t \geq 0$ and $x'_2 = 0$. Because the last two columns form the identity matrix, these choices determine x'_3 and x'_4. We have:

$$\begin{aligned}
\begin{bmatrix} 8 \\ 10 \end{bmatrix} &= x'_1 \begin{bmatrix} 2 \\ 2 \end{bmatrix} + x'_2 \begin{bmatrix} 2 \\ 1 \end{bmatrix} + x'_3 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + x'_4 \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
&= t \begin{bmatrix} 2 \\ 2 \end{bmatrix} + \begin{bmatrix} x'_3 \\ x'_4 \end{bmatrix} \\
\begin{bmatrix} x'_3 \\ x'_4 \end{bmatrix} &= \begin{bmatrix} 8 \\ 10 \end{bmatrix} - t \begin{bmatrix} 2 \\ 2 \end{bmatrix}.
\end{aligned}$$

Thus $x'_3 = 8 - 2t$ and $x'_4 = 10 - 2t$. Nonnegativity requires $t \leq 4$ and $t \leq 5$, so the largest admissible value is $t = 4$. This gives the improved feasible solution

$$x' = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 2 \end{bmatrix}, \qquad z = 16.$$

Lecture 8 — Thursday, January 30, 2014

Consider a closely related pivot calculation.

Example 8.1

$$\begin{aligned} \max \quad & z = \begin{bmatrix} 4 \\ 5 \\ 0 \\ 0 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \underbrace{\begin{bmatrix} 4 & 1 & 1 & 0 \\ 2 & 2 & 0 & 1 \end{bmatrix}}_{=A} \vec{x} = \begin{bmatrix} 8 \\ 10 \end{bmatrix}, \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

We want to improve the feasible solution $\vec{x} = \begin{bmatrix} 0 \\ 0 \\ 8 \\ 10 \end{bmatrix}$.

To find a better feasible solution $\vec{x}'$ with $z > 0$, set $x_1' = t \geq 0$ and $x_2' = 0$. Then

$$\begin{bmatrix} 8 \\ 10 \end{bmatrix} = x_1' \begin{bmatrix} 4 \\ 2 \end{bmatrix} + x_2' \begin{bmatrix} 1 \\ 2 \end{bmatrix} + x_3' \begin{bmatrix} 1 \\ 0 \end{bmatrix} + x_4' \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Then $\begin{bmatrix} x_3' \\ x_4' \end{bmatrix} = \begin{bmatrix} 8 \\ 10 \end{bmatrix} - t \begin{bmatrix} 4 \\ 2 \end{bmatrix}$.

Then we get $x_1' = t \geq 0$ and $x_2' = 0$, while $x_3' = 8 - 4t \geq 0$ gives $t \leq \frac{8}{4} = 2$ and $x_4' = 10 - 2t \geq 0$ gives $t \leq \frac{10}{2} = 5$.

Therefore the largest admissible value is $t = \min\{2, 5\} = 2$.

So we get $\vec{x}' = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 6 \end{bmatrix}$, and $z = \begin{bmatrix} 4 \\ 5 \\ 0 \\ 0 \end{bmatrix} \cdot x' = 8$.

Can we continue by increasing x_2 from this new solution? If we freeze $x_1'' = 2$ and set $x_2'' = t \geq 0$, the first equation gives $x_3'' = -t$, so this particular move immediately violates nonnegativity. That does not mean improvement is impossible: we must let the basic variables move together. Keeping $x_3'' = 0$ and setting $x_2'' = t$ gives

$$x_1'' = 2 - \frac{t}{4}, \qquad x_4'' = 6 - \frac{3t}{2}, \qquad 0 \leq t \leq 4.$$

Along this feasible move, $z = 4x_1'' + 5x_2'' = 8 + 4t$. Taking $t = 4$ yields $\vec{x}'' = \begin{bmatrix} 1 & 4 & 0 & 0 \end{bmatrix}^\top$ and $z = 24$. Systematic basis pivots keep track of precisely these coordinated changes.

The first step in Example 8.1 was easy because the last two columns of A form the identity matrix $I_{2\times 2}$ and the corresponding coefficients in the objective are zero. This motivates the formal theory of bases.

Notation 8.2 Let A be an $m \times n$ matrix, let $J \subseteq \{1, \ldots, n\}$ be a set of columns. Then A_J is defined as the matrix obtained by selecting columns indexed by J. By convention, we take columns in the same order.

Example 8.3 Let

$$A = \begin{bmatrix} 1 & 2 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 1 & 0 & -1 \\ 1 & 3 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

If $J = \{1, 2, 4\}$, then

$$A_J = \begin{bmatrix} 1 & 2 & 0 \\ -1 & 1 & 1 \\ 1 & 3 & 0 \end{bmatrix}.$$

If $|J| = 1$, then, for example, $A_2 = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}$.

Definition 8.4 $B \subseteq \{1, \ldots, n\}$ is a **basis** if A_B is both square and invertible.

Example 8.5 Again let

$$A = \begin{bmatrix} 1 & 2 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 1 & 0 & -1 \\ 1 & 3 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Then $A_{\{3,5\}}$ is not square. $A_{\{4,5,6\}}$ is not invertible. $A_{\{3,4,5\}}$ is a basis. $A_{\{1,4,5\}}$ is also a basis.

We will assume that the rows of A are linearly independent. Then we can redefine what a basis is:

Proposition 8.6 B is a basis if and only if A_B is a maximum set of linearly independent columns of A.

This matters because the simplex algorithm works with special solutions associated with bases for systems in SEF. Can we assume that the rows of A are independent? Yes. Row reduction either produces an inconsistent row, which certifies infeasibility, or identifies dependent equations that are redundant and may be removed without changing the feasible set. We may therefore work with a full-row-rank system.

Basic Solutions

The simplex method is organized around these basis-dependent solutions, so we now define them carefully.

Say $A\vec{x} = \vec{b}$, assuming the rows of A are independent and A is $m \times n$. Let B be a basis of A. Let $N := \{1, \ldots, n\} \setminus B$.

Definition 8.7 x_j is a **basic variable** if $j \in B$. x_j is a **nonbasic variable** if $j \in N$.

Example 8.8 Consider

$$\begin{bmatrix} 1 & 2 & 0 & -1 \\ -1 & 3 & 1 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}.$$

Pick, for instance, $B = \{1, 3\}$ which is a basis. The basic variables are x_1, x_3, and the nonbasic ones are x_2, x_4, and

$$A_B = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \qquad A_N = \begin{bmatrix} 2 & -1 \\ 3 & 1 \end{bmatrix}, \qquad \vec{x}_B = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}, \qquad \text{and} \qquad \vec{x}_N = \begin{bmatrix} x_2 \\ x_4 \end{bmatrix}.$$

(Check that $A\vec{x} = A_B\vec{x}_B + A_N\vec{x}_N$).

This works in general.

Proposition 8.9 For any vector $\vec{x} \in \mathbb{R}^n$, we have

$$A\vec{x} = A_B\vec{x}_B + A_N\vec{x}_N.$$

Proof.

$$A\vec{x} = \sum_{j=1}^{n} x_j A_j = \sum_{j\in B} x_j A_j + \sum_{j\in N} x_j A_j = A_B\vec{x}_B + A_N\vec{x}_N.$$

□

Definition 8.10 Given $A\vec{x} = \vec{b}$ and a basis B, we say that $\bar{\vec{x}}$ is the **basic solution** for B if

(1) $A\bar{\vec{x}} = \vec{b}$, and

(2) $\bar{\vec{x}}_N = \vec{0}$ (that is, $\bar{x}_j = 0$ for all $j \in N$).

The use of the phrase *the* basic solution (rather than *a* basic solution) in Definition 8.10 will be justified shortly.

Example 8.11 Again consider

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

and the basis $B = \{1, 3\}$. Then

$$\bar{\vec{x}} = \begin{bmatrix} 1 \\ 0 \\ 5 \\ 0 \end{bmatrix}$$

is the basic solution for B.

Proposition 8.12 For $A\vec{x} = \vec{b}$ and a basis B, there exists a unique basic solution for B.

Proof. Since B is a basis, A_B is invertible.

Define

$$\bar{\vec{x}}_N := \vec{0} \qquad \text{and} \qquad \bar{\vec{x}}_B := A_B^{-1}\vec{b}.$$

Then

$$A\bar{\vec{x}} = A_B\bar{\vec{x}}_B + A_N\bar{\vec{x}}_N = A_B(A_B^{-1}\vec{b}) + A_N\vec{0} = \vec{b}.$$

So $\bar{\vec{x}}$ is a basic solution for B, and existence holds.

For uniqueness, let $\vec{x}$ be any basic solution for B. Then $\vec{x}_N = \vec{0}$ and

$$\vec{b} = A\vec{x} = A_B\vec{x}_B + A_N\vec{x}_N = A_B\vec{x}_B.$$

Since A_B is invertible, $\vec{x}_B = A_B^{-1}\vec{b} = \bar{\vec{x}}_B$, and together with $\vec{x}_N = \bar{\vec{x}}_N = \vec{0}$ this gives $\vec{x} = \bar{\vec{x}}$. Therefore the basic solution for B is unique. $\square$

Lecture 9 — Tuesday, February 04, 2014

Canonical Form and Pivoting

With the basic simplex ideas in place, we now study canonical form as the language for systematic pivot operations.

The following quick checks build intuition for which vectors can arise as basic solutions.

1. Recall the system $\begin{bmatrix} -1 & 1 & 2 & 1 \\ 1 & -1 & 1 & 0 \end{bmatrix} \vec{x} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$. To show that $B = \{1,3\}$ is a basis, it suffices to check that $A_B = \begin{bmatrix} -1 & 2 \\ 1 & 1 \end{bmatrix}$ is nonsingular.

2. The basic solution for $B = \{1,3\}$ is $\vec{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^\top$. It is feasible and degenerate. Because only x_3 is positive, the same vector is also the basic solution for the bases $\{2,3\}$ and $\{3,4\}$.

3. The vector $\vec{x} = \begin{bmatrix} 1 & 0 & 0 & 3 \end{bmatrix}^\top$ is the basic solution for $B = \{1,4\}$ and is feasible.

4. The vector $\vec{x} = \begin{bmatrix} 0 & -1 & 0 & 3 \end{bmatrix}^\top$ is the basic solution for $B = \{2,4\}$, but it is not feasible because $x_2 < 0$.

5. The vector $\vec{x} = \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix}^\top$ is feasible but is not basic for any basis: it has three positive entries, whereas every basis contains only two columns.

Example 9.1

$$\begin{aligned} \max \quad & z = \begin{bmatrix} 4 \\ 5 \\ 0 \\ 0 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 2 & 1 & 1 & 0 \\ 2 & 2 & 0 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 8 \\ 10 \end{bmatrix}. \end{aligned}$$

One feasible solution is $\vec{x} = \begin{bmatrix} 0 \\ 0 \\ 8 \\ 10 \end{bmatrix}$.

Definition 9.2 The LP

$$\begin{aligned} \max \quad & z = \vec{c}^\top \vec{x} + \bar{z} \\ \text{subject to} \quad & A\vec{x} = \vec{b}, \\ & \vec{x} \geq \vec{0} \end{aligned}$$

(where $\bar{z}$ is a constant) is in **canonical form** for some basis B if:

(1) $A_B = I$ and

(2) $\vec{c}_B = \vec{0}$.

Proposition 9.3 For any basis B, we can rewrite the LP

$$\begin{aligned} \max \quad & z = \vec{c}^\top \vec{x} \\ \text{subject to} \quad & A\vec{x} = \vec{b}, \\ & \vec{x} \geq \vec{0} \end{aligned}$$

in canonical form for B so that

(1) we do not change the set of feasible solutions, and

(2) for any feasible solution, we have the same value for the objective function.

Proof. First rewrite the constraints by multiplying on the left by A_B^{-1}:

$$A\vec{x} = \vec{b} \iff A_B^{-1}A\vec{x} = A_B^{-1}\vec{b}.$$

This equivalence holds because A_B^{-1} is invertible. Therefore the feasible set is unchanged, proving ((1)).

In the transformed system, the basic block is

$$(A_B^{-1}A)_B = A_B^{-1}A_B = I,$$

so condition ((1)) in the definition of canonical form is satisfied.

Now adjust the objective so that condition ((2)) holds while preserving objective values on feasible points. Write

$$z = \vec{c}_B^\top \vec{x}_B + \vec{c}_N^\top \vec{x}_N.$$

From $A_B^{-1}A\vec{x} = A_B^{-1}\vec{b}$ we get

$$\vec{x}_B + (A_B^{-1}A_N)\vec{x}_N = A_B^{-1}\vec{b},$$

and hence

$$\vec{x}_B = A_B^{-1}\vec{b} - A_B^{-1}A_N\vec{x}_N.$$

Substitute into z to get

$$z = \vec{c}_B^\top A_B^{-1}\vec{b} + \big(\vec{c}_N^\top - \vec{c}_B^\top A_B^{-1}A_N\big)\vec{x}_N.$$

Therefore we can define

$$\bar{z} := \vec{c}_B^\top A_B^{-1}\vec{b}, \qquad \vec{\bar{c}}_B := \vec{0}, \qquad \text{and} \qquad \vec{\bar{c}}_N := \vec{c}_N - A_N^\top (A_B^{-\top}\vec{c}_B),$$

and write the equivalent objective in canonical form as

$$z = \bar{z} + \vec{\bar{c}}^\top \vec{x}.$$

This expression equals the original objective value for every feasible $\vec{x}$, so ((2)) follows. □

Example 9.4

$$\begin{aligned} \max \quad & z = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} -1 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

Suppose $B = \{1, 2\}$ is a basis. We want to rewrite the LP in canonical form.

To force $A_B = I$, multiply the constraint system on the left by A_B^{-1}:

$$\begin{bmatrix} -1 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} -1 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} -1 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 2 \\ 1 \end{bmatrix}.$$

Hence

$$\frac{1}{-2-1} \begin{bmatrix} 2 & -1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} -1 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 1 & 0 & -\frac{1}{3} \\ 0 & 1 & \frac{2}{3} \end{bmatrix} \vec{x} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

We also want the coefficients of the basic variables in the objective to be zero. Since $\vec{b} - A\vec{x} = \vec{0}$, multiply this equation by $\begin{bmatrix} 0 \\ 1 \end{bmatrix}^\top$ to obtain

$$1 - \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \cdot \vec{x} = 0.$$

Adding this identity to the objective gives

$$z = 1 + \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} \cdot \vec{x},$$

so the coefficients of the basic variables are now zero.

In general, replace $A\vec{x} = \vec{b}$ by $A_B^{-1}A\vec{x} = A_B^{-1}\vec{b}$.

Proposition 9.5

(1) If $\bar{\vec{x}}$ is a solution to $A\vec{x} = \vec{b}$, it is a solution to $A_B^{-1}A\vec{x} = A_B^{-1}\vec{b}$.

(2) If $\bar{\vec{x}}$ is a solution to $A_B^{-1}A\vec{x} = A_B^{-1}\vec{b}$, then it is a solution to $A\vec{x} = \vec{b}$.

The proof is on Assignment 1.

In general, we proceed in two steps. First construct a new identity $\vec{y}^\top\vec{b} - \vec{y}^\top A\vec{x} = 0$ from $\vec{b} - A\vec{x} = \vec{0}$. Then add this identity to the objective function to obtain $\vec{y}^\top\vec{b} - (\vec{y}^\top A - \vec{c}^\top)\vec{x} = z$. Any choice of $\vec{y}$ leaves the LP unchanged, so we choose $\vec{y}$ to make the basic coefficients vanish. Since $\bar{\vec{c}}^\top = \vec{c}^\top - \vec{y}^\top A$, the condition $\bar{\vec{c}}_B^\top = \vec{0}^\top$ becomes

$$\vec{c}_B^\top - \vec{y}^\top A_B = \vec{0}^\top,$$

which is equivalent to $\vec{y}^\top A_B = \vec{c}_B^\top$. Equivalently, $A_B^\top\vec{y} = \vec{c}_B$, so

$$\vec{y} = (A_B^\top)^{-1}\vec{c}_B = A_B^{-\top}\vec{c}_B.$$

Thus we have proven the following:

Proposition 9.6 Consider the LP

$$\begin{aligned} \max \quad & z = \vec{c}^\top\vec{x} \\ \text{subject to} \quad & A\vec{x} = \vec{b}, \\ & \vec{x} \geq \vec{0} \end{aligned}$$

and suppose that B is a basis of A. The following equivalent LP is in canonical form for B:

$$\begin{aligned} \max \quad & z = \vec{b}^\top \vec{y} + (\vec{c}^\top - \vec{y}^\top A)\vec{x} \\ \text{subject to} \quad & A_B^{-1} A\vec{x} = A_B^{-1}\vec{b}, \\ & \vec{x} \geq \vec{0}, \\ & \vec{y} = A_B^{-\top}\vec{c}_B \end{aligned}$$

Example 9.7 We now run the simplex algorithm in canonical form on the same LP as our running pivot example:

$$\begin{aligned} \max \quad & z = \begin{bmatrix} 4 \\ 5 \\ 0 \\ 0 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 2 & 1 & 1 & 0 \\ 2 & 2 & 0 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 8 \\ 10 \end{bmatrix}, \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

Start with basis $B = \{1, 4\}$, so the corresponding basic feasible solution is

$$\vec{x} = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 2 \end{bmatrix}.$$

Rewriting in canonical form for $B = \{1, 4\}$ gives

$$\begin{aligned} \max \quad & z = 16 + \begin{bmatrix} 0 \\ 3 \\ -2 \\ 0 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 1 & 1/2 & 1/2 & 0 \\ 0 & 1 & -1 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}, \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

This LP is in canonical form and is equivalent to the original LP, so it is a better starting point for the algorithm.

Set $x_2 = t$ and $x_3 = 0$. Then

$$\begin{bmatrix}1\\0\end{bmatrix} x'_1 + \begin{bmatrix}1/2\\1\end{bmatrix} x'_2 + \begin{bmatrix}1/2\\-1\end{bmatrix} x'_3 + \begin{bmatrix}0\\1\end{bmatrix} x'_4 = \begin{bmatrix}4\\2\end{bmatrix}.$$

Rearranging gives

$$\begin{bmatrix}x'_1\\x'_4\end{bmatrix} = \begin{bmatrix}4\\2\end{bmatrix} - t\begin{bmatrix}1/2\\1\end{bmatrix} \geq \begin{bmatrix}0\\0\end{bmatrix}.$$

The first component gives $4 - t/2 \geq 0$, or $t \leq 8$, while the second gives $2 - t \geq 0$, or $t \leq 2$. The binding restriction is therefore $t \leq 2$, and we choose $t = 2$.

The new solution is $x' = \begin{bmatrix}3\\2\\0\\0\end{bmatrix}$. This is a basic solution for the new basis $B' = \{1, 2\}$.

Rewriting in canonical form for $B' = \{1, 2\}$ gives

$$\begin{aligned} \max \quad & z = 22 + \begin{bmatrix}0\\0\\1\\-3\end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix}1 & 0 & 1 & -1/2\\0 & 1 & -1 & 1\end{bmatrix} \vec{x} = \begin{bmatrix}3\\2\end{bmatrix}, \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

Lecture 10 — Thursday, February 06, 2014

Example 10.1 (Continuation of the simplex pivot example) For the second iteration, choose the entering variable x_3 (its reduced cost is $+1$), and set $x_4 = 0$, $x_3 = t$. Then

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix} - t \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 3-t \\ 2+t \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

so $t \leq 3$. Taking $t = 3$ gives the new basis $\{2, 3\}$ and basic feasible solution

$$\vec{x} = \begin{bmatrix} 0 \\ 5 \\ 3 \\ 0 \end{bmatrix}.$$

In canonical form for $\{2, 3\}$, the LP is

$$\begin{aligned} \max \quad & z = 25 + \begin{bmatrix} -1 \\ 0 \\ 0 \\ -5/2 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 1 & 1 & 0 & 1/2 \\ 1 & 0 & 1 & -1/2 \end{bmatrix} \vec{x} = \begin{bmatrix} 5 \\ 3 \end{bmatrix}, \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

All reduced costs of nonbasic variables are now nonpositive, so the algorithm terminates after two iterations. We now certify optimality.

Proposition 10.2 25 is an upper bound for the LP.

Proof. Let $\bar{\vec{x}}$ be any feasible solution. Then

$$z = 25 + \begin{bmatrix} -1 \\ 0 \\ 0 \\ -5/2 \end{bmatrix} \cdot \bar{\vec{x}}.$$

Since $\begin{bmatrix} -1 \\ 0 \\ 0 \\ -5/2 \end{bmatrix} \leq \vec{0}$ and $\bar{\vec{x}} \geq \vec{0}$, the correction term is at most 0. Hence 25 is an upper bound.

Because $\vec{x} = \begin{bmatrix} 0 \\ 5 \\ 3 \\ 0 \end{bmatrix}$ attains this value, it is optimal. □

Canonical form also makes unboundedness transparent, so we examine a representative case.

Example 10.3

$$\begin{aligned} \max \quad & z = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 2 \\ 1 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 1 & 0 & 2 & -1 & 1 \\ 0 & 1 & 1 & -3 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 6 \\ 5 \end{bmatrix}, \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

This LP is in canonical form for $B = \{1, 2\}$ with $\vec{x} = \begin{bmatrix} 6 \\ 5 \\ 0 \\ 0 \\ 0 \end{bmatrix}$. We work with x_4 by setting $x_4 = t$ and $x_3 = x_5 = 0$. Then

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \end{bmatrix} - t \begin{bmatrix} -1 \\ -3 \end{bmatrix} = \begin{bmatrix} 6 + t \\ 5 + 3t \end{bmatrix}.$$

We may choose t as large as we want, so the LP is unbounded. If we had started with x_5 instead, we would eventually reach the same conclusion. Our certificate of unboundedness is

$$\vec{x} = \begin{bmatrix} 6+t \\ 5+3t \\ 0 \\ t \\ 0 \end{bmatrix} = \underbrace{\begin{bmatrix} 6 \\ 5 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\bar{\vec{x}}} + t \underbrace{\begin{bmatrix} 1 \\ 3 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\vec{d}}.$$

This example shows how unboundedness appears naturally in canonical form.

With optimality and unboundedness criteria in place, we can now state the simplex algorithm formally.

Algorithm 1: The Simplex Algorithm

Input: A linear program $\max \vec{c}^\top \vec{x} + \bar{z}$ subject to $A\vec{x} = \vec{b}$, $\vec{x} \geq \vec{0}$, together with a feasible basis B

Output: An optimal basic feasible solution with certificate of optimality, or a certificate that the LP is unbounded

Rewrite the LP in canonical form for basis B;
if $\vec{c}_N \leq \vec{0}$ **then**
  return *the current basic feasible solution (optimal; the vector* $\vec{y} = A_B^{-\top}\vec{c}_B$*, computed from the original basis matrix and objective coefficients, certifies it)*;
end
Choose $k \in N$ with $c_k > 0$;
if $A_{\cdot k} \leq 0$ **then**
  return *the LP is unbounded;*
  ;
end
Set $t \leftarrow \min\{b_i/A_{ik} \mid A_{ik} > 0\}$;
Let r be the row index attaining this minimum, and let l be the r-th basic variable;
Set $B' \leftarrow (B \cup \{k\}) \setminus \{l\}$ and repeat from the first step with basis B';

For

$$\begin{aligned} \max \quad & z = \vec{c}^\top \vec{x} \qquad (P) \\ \text{subject to} \quad & A\vec{x} = \vec{b}, \\ & \vec{x} \geq \vec{0}, \end{aligned}$$

the following table summarizes our current situation:

Task	
A Solve (P)	What we want to do
B Find a feasible solution for (P)	?
C Given a feasible solution for (P), solve (P)	What we know how to do

Finding a feasible solution is not easy. In fact, it is computationally as difficult as finding an optimal solution. The next lecture addresses this missing step.

Lecture 11 — Tuesday, February 11, 2014

Two-Phase Simplex and Feasibility

A central practical issue is how to obtain a feasible starting basis, and the two-phase simplex algorithm addresses exactly this point.

Before optimization begins, we must find a feasible starting point. To this end, consider the problem of finding a feasible solution to

$$\begin{aligned} \max \quad & z = \dots & \quad (P) \\ \text{subject to} \quad & \begin{bmatrix} 1 & 2 & 1 \\ -3 & -1 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 3 \\ -4 \end{bmatrix} \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

1) First eliminate the negative entry of $\vec{b}$:

$$\begin{bmatrix} 1 & 2 & 1 \\ 3 & 1 & -1 \end{bmatrix} \vec{x} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

2) Then introduce new variables x_4 and x_5 to create an identity block:

$$\left.\begin{array}{rl} \min & x_4 + x_5 \qquad\qquad\qquad (Q) \\ \text{subject to} & \begin{bmatrix} 1 & 2 & 1 & 1 & 0 \\ 3 & 1 & -1 & 0 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \\ & \vec{x} \geq \vec{0} \end{array}\right\} \text{auxiliary problem}$$

3) A convenient feasible solution is $\bar{\vec{x}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3 \\ 4 \end{bmatrix}$.

4) Solving (Q) using this feasible solution gives the optimal solution $\vec{x}' = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$.

Therefore $\vec{x}'' = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ is feasible for (P) after we drop the auxiliary variables.

Example 11.1 Now consider the problem of finding a feasible solution to

$$\begin{array}{rl} \max & z = \dots \\ \text{subject to} & \begin{bmatrix} 1 & 2 & 2 \\ 2 & 2 & 3 \end{bmatrix} \vec{x} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \\ & \vec{x} \geq \vec{0} \end{array}$$

The auxiliary model packages the feasibility search as an optimization problem.

$$\begin{array}{rl} \min & x_4 + x_5 \qquad\qquad\qquad (Q) \\ \text{subject to} & \begin{bmatrix} 1 & 2 & 2 & 1 & 0 \\ 2 & 2 & 3 & 0 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \\ & x_1, \dots, x_5 \geq 0 \end{array}$$

One feasible solution is $\bar{\vec{x}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2 \\ 1 \end{bmatrix}$, and from this an optimal solution is $\vec{x}' = \begin{bmatrix} 0 \\ 1/2 \\ 0 \\ 1 \\ 0 \end{bmatrix}$, with objective value 1.

Therefore we cannot recover a feasible solution to (P) in the same way as before, and

Proposition 11.2 (P) has no feasible solution.

Proof. Suppose $\begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix}$ were a feasible solution to (P). Then $\begin{bmatrix} x_1' \\ x_2' \\ x_3' \\ 0 \\ 0 \end{bmatrix}$ is a feasible solution to (Q) that has $w = 0$. But the optimal solution to (Q) has value 1. This is a contradiction. Therefore (P) has no feasible solution. ☐

We now summarize the full two-phase framework in a compact template. Start with (P):

$$\begin{aligned} \max \quad & \vec{c}^\top \vec{x} & \qquad (P) \\ \text{subject to} \quad & A\vec{x} = \vec{b} \\ & \vec{x} \geq \vec{0} \end{aligned}$$

where A is an $m \times n$ matrix and $\vec{b} \geq \vec{0}$.

Phase 1 First form the auxiliary program (Q):

$$\begin{aligned} \min \quad & w = x_{n+1} + \cdots + x_{n+m} & \qquad (Q) \\ \text{subject to} \quad & \begin{bmatrix} A & \mid & I \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{n+m} \end{bmatrix} = \vec{b} \end{aligned}$$

$$x_1, \ldots, x_{n+m} \geq 0$$

Then pick the feasible basic solution $\vec{x}' = \begin{bmatrix} \vec{0} \\ \vec{b} \end{bmatrix}$. Put (Q) in SEF form and run the simplex algorithm to find an optimal solution $\vec{x}''$.

If the optimal auxiliary value satisfies $w^* > 0$, then stop: (P) is infeasible. Otherwise, $w^* = 0$, and dropping the artificial variables from $\vec{x}''$ gives a feasible solution to (P). If a zero-valued artificial variable remains basic, pivot it out; if that is impossible, remove the corresponding redundant row. This produces a feasible basis for the original variables.

Phase 2 Once we have a feasible basis for (P), the second phase is exactly the ordinary simplex algorithm applied to the original objective.

We still need guarantees on correctness and finite progress. If the algorithm gives an answer, that answer is correct. However, the simplex algorithm may fail to terminate because of **cycling**.

To guarantee termination, use **Bland's rule**: whenever there is a choice of entering or leaving variables, choose the eligible variable with the smallest index.

Example 11.3

$$\begin{aligned} \max \quad & \begin{bmatrix} -3 \\ 6 \\ 1 \\ 0 \\ 0 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 1 & 2 & 1 & 1 & 0 \\ 2 & 4 & 3 & 0 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 8 \\ 7 \end{bmatrix} \\ & \vec{x} \geq \vec{0} \end{aligned}$$

We may choose either x_2 or x_3 to enter the basis since $c_2, c_3 > 0$. We select x_2 because it has the smaller index.

These structural consequences explain why two-phase simplex is foundational for linear programming theory. We already stated the following theorem earlier in the course; our development of the two-phase simplex method has now proven it.

Theorem 11.4 (Fundamental theorem of linear programming) For the linear program

$$\begin{array}{rll} \max & \vec{c}^{\top}\vec{x} & (P) \\ \text{subject to} & A\vec{x} = \vec{b} & \\ & \vec{x} \geq \vec{0} & \end{array}$$

exactly one of the following holds:

1. (P) is infeasible.
2. (P) is unbounded.
3. (P) has an optimal solution.

The following theorem has also been established by the two-phase simplex method:

Theorem 11.5 If (P) has an optimum solution, then it has an optimum solution that is a basic solution.

3. Geometry

We now interpret linear programming through geometry, which explains why methods based on extreme points are effective.

Feasible Regions: Hyperplanes and Half-Spaces

We begin the geometric viewpoint by naming the feasible region and asking what its shape tells us algorithmically.

Definition 11.6 The **feasible region** of an optimization problem is the set of all feasible solutions.

This raises three natural questions: What is the shape of the feasible region of an LP? What do extreme points correspond to? Why does this make LPs easier to solve?

Example 11.7

$$\begin{array}{rl} \max & \begin{bmatrix} 2 \\ 1 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} & \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \vec{x} \leq \begin{bmatrix} 3 \\ 2 \\ 2 \\ 0 \\ 0 \end{bmatrix} \begin{array}{c} ① \\ ② \\ ③ \\ ④ \\ ⑤ \end{array} \end{array}$$

The feasible region in Figure 12 is a polyhedron.

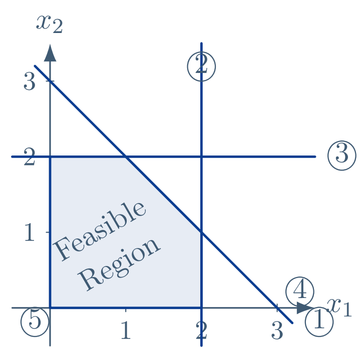


FIGURE 12

Let $\vec{\alpha} \in \mathbb{R}^n$ with $\vec{\alpha} \neq \vec{0}$, and let $\beta \in \mathbb{R}$.

Definition 11.8 The set $H = \{\vec{x} \mid \vec{\alpha}^\top \vec{x} = \beta\}$ is called a **hyperplane**. The set $F = \{\vec{x} \mid \vec{\alpha}^\top \vec{x} \leq \beta\}$ is called a **half-space**.

The situation is easy to visualize in $\mathbb{R}^2$ when $\beta = 0$, as shown in Figure 13.

From Figure 13, we can see that H is the set of vectors perpendicular to $\vec{\alpha}$, and the nonzero points of F are precisely those that form an angle of at least $90°$ with $\vec{\alpha}$.

Proposition 11.9 The affine dimension of H is $n - 1$.

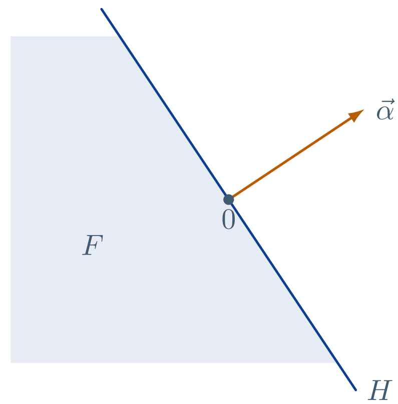


Figure 13

Proof. Recall from linear algebra that for an $m \times n$ matrix A, the rank–nullity theorem states that

$$\dim\{\vec{x} \mid A\vec{x} = \vec{0}\} + \operatorname{rank}(A) = n.$$

Replacing A with $\vec{\alpha}^\top$,

$$\dim\{\vec{x} \mid \vec{\alpha}^\top\vec{x} = 0\} + \operatorname{rank}(\vec{\alpha}^\top) = n,$$

and since $\vec{\alpha}^\top$ is a rank-1 matrix, this kernel has dimension $n-1$. The hyperplane H is a translate of the kernel, so it has the same affine dimension. □

Lecture 12 — Thursday, February 13, 2014

Building on the previous geometric view, we now formalize polyhedra and their structural properties.

Polyhedra

A **translated hyperplane** is the natural next step after studying homogeneous hyperplanes. Consider adding a constant vector $\bar{\vec{x}}$ to every element in a homogeneous hyperplane H. The resulting set H'

$$H' = \{\vec{x} + \bar{\vec{x}} \mid \vec{\alpha}^\top\vec{x} = 0\} = \{\vec{x}' \mid \vec{\alpha}^\top(\vec{x}' - \bar{\vec{x}}) = 0\} = \{\vec{x}' \mid \vec{\alpha}^\top\vec{x}' = \beta\}$$

where $\beta := \vec{\alpha}^\top\bar{\vec{x}}$.

Figure 14 shows one translation from H to H'.

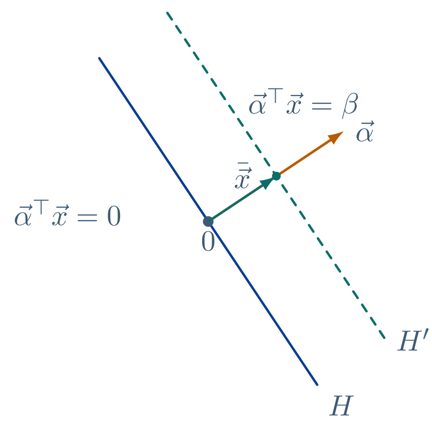


FIGURE 14

The feasible region of an LP is of the form $\{\vec{x} \mid A\vec{x} \leq \vec{b}\}$ for some A and $\vec{b}$. Such a set is the intersection of finitely many half-spaces, so it is a polyhedron.

Definition 12.1 A **polyhedron** is an intersection of finitely many half-spaces.

Figures 15 to 17 contrast a polyhedron with two sets that are not polyhedra.

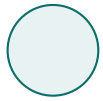
FIGURE 15

FIGURE 16

Convexity

Convexity is the geometric property that makes local improvement arguments globally meaningful.

Definition 12.2 Let $\vec{x}^1, \vec{x}^2 \in \mathbb{R}^n$. The **line segment** between $\vec{x}^1$ and $\vec{x}^2$ is $\{\vec{x} \in \mathbb{R}^n \mid \vec{x} = \lambda\vec{x}^1 + (1-\lambda)\vec{x}^2, \lambda \in [0,1]\}$.



This is not a polyhedron.

FIGURE 17

Definition 12.3 $C \subseteq \mathbb{R}^n$ is **convex** if for all $\vec{x}^1, \vec{x}^2 \in C$, the line segment between them is still in C.

Proposition 12.4 Let C_1 and C_2 be convex. In general, $C_1 \cup C_2$ need not be convex, but $C_1 \cap C_2$ is convex.

Proof. For the union claim, a counterexample suffices. In $\mathbb{R}$, let

$$C_1 = [-2, -1] \qquad \text{and} \qquad C_2 = [1, 2].$$

Each set is convex, but $C_1 \cup C_2$ is not convex because $-1, 1 \in C_1 \cup C_2$ while their midpoint

$$\frac{-1+1}{2} = 0$$

is not in $C_1 \cup C_2$.

Now we prove $C_1 \cap C_2$ is convex. Take any $\vec{x}^1, \vec{x}^2 \in C_1 \cap C_2$ and any $\lambda \in [0, 1]$. Since $\vec{x}^1, \vec{x}^2 \in C_1$ and C_1 is convex,

$$\lambda \vec{x}^1 + (1 - \lambda)\vec{x}^2 \in C_1.$$

Likewise, since $\vec{x}^1, \vec{x}^2 \in C_2$ and C_2 is convex,

$$\lambda \vec{x}^1 + (1 - \lambda)\vec{x}^2 \in C_2.$$

Therefore

$$\lambda \vec{x}^1 + (1 - \lambda)\vec{x}^2 \in C_1 \cap C_2.$$

Hence $C_1 \cap C_2$ is convex. □

Theorem 12.5 Any polyhedron P is convex.

Proof. Let $P = \{\vec{x} \mid A\vec{x} \leq \vec{b}\}$ for some $\vec{b}$ and A.

Pick $\vec{x}^1, \vec{x}^2 \in P$, and pick $\bar{\vec{x}}$ in the line segment between $\vec{x}^1$ and $\vec{x}^2$. We want to show that $\bar{\vec{x}} \in P$. Since $\vec{x}^1, \vec{x}^2 \in P$, we have $A\vec{x}^1 \leq \vec{b}$ and $A\vec{x}^2 \leq \vec{b}$. $\bar{\vec{x}} = \lambda\vec{x}^1 + (1-\lambda)\vec{x}^2$ where $0 \leq \lambda \leq 1$. We check that $A\bar{\vec{x}} \leq \vec{b}$:

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

$\square$

Extreme Points and Basic Feasible Solutions

Extreme points capture where optimal solutions tend to occur, so we define them carefully.

Definition 12.6 Let $\vec{x}^1, \vec{x}^2 \in \mathbb{R}^n$ and let L be the line segment between $\vec{x}^1$ and $\vec{x}^2$. Then L **properly contains** $\bar{\vec{x}}$ if $\bar{\vec{x}} \in L$ and $\bar{\vec{x}} \notin \{\vec{x}^1, \vec{x}^2\}$.

Definition 12.7 Let $C \subseteq \mathbb{R}^n$ be convex. A point $\vec{x} \in C$ is **not extreme** if there exists a line segment in C that properly contains $\vec{x}$. Otherwise, $\vec{x}$ is an **extreme point** of C.

Equivalently, $\vec{x}$ *is* an extreme point of C if and only if there is no line segment contained in C that strictly contains $\vec{x}$.

Every vertex of a polyhedron is an extreme point. By contrast, every boundary point of a closed disk is extreme, even though the disk is not a polyhedron.

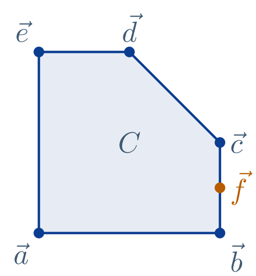


FIGURE 18

In Figure 18, $\vec{a}, \ldots, \vec{e}$ are the extreme points of C. The point $\vec{f}$ is not extreme because the line segment between $\vec{b}$ and $\vec{c}$ properly contains it.

Definition 12.8 A linear constraint $\vec{\alpha}^\top \vec{x} \leq \beta$ is **tight** for a feasible vector $\bar{\vec{x}}$ if the constraint is satisfied with equality; i.e., $\vec{\alpha}^\top \bar{\vec{x}} = \beta$.

Notation 12.9 Let $P = \{\vec{x} \mid A\vec{x} \leq \vec{b}\} \subseteq \mathbb{R}^n$ be a polyhedron. Let $\bar{\vec{x}} \in P$. Denote by $A^=$ the matrix formed by the rows of A corresponding to tight constraints for $\bar{\vec{x}}$, and define $\vec{b}^=$ similarly, so that $A^=\bar{\vec{x}} = \vec{b}^=$.

Example 12.10 Consider

$$P = \left\{ \vec{x} \in \mathbb{R}^2 \;\middle|\; \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \vec{x} \leq \begin{bmatrix} 3 \\ 2 \\ 2 \\ 0 \\ 0 \end{bmatrix} \right\}.$$

Figure 19 marks one extreme point and one nonextreme boundary point of P.

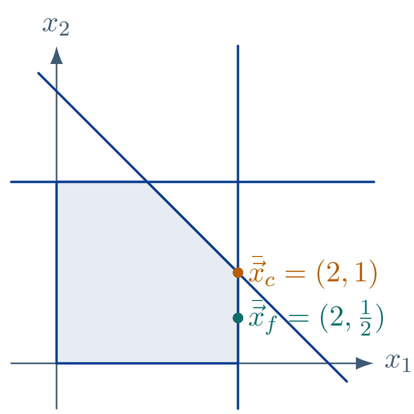


FIGURE 19

At $\bar{\vec{x}}_c = (2, 1)$, the tight constraints are $x_1 + x_2 = 3$ and $x_1 = 2$, so

$$A_c^= = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \qquad \text{and} \qquad \operatorname{rank}(A_c^=) = 2 = n.$$

Hence $\bar{\vec{x}}_c$ is an extreme point.

At $\bar{\vec{x}}_f = (2, \frac{1}{2})$, the only tight constraint is $x_1 = 2$, so

$$A_f^= = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad \text{and} \qquad \operatorname{rank}(A_f^=) = 1 < 2 = n.$$

Hence $\bar{\vec{x}}_f$ is not an extreme point.

Lecture 13 — Tuesday, February 25, 2014

We now connect the geometric notion of extremity to the algebraic notion of a basic feasible solution.

Theorem 13.1 Let $P = \{\vec{x} \mid A\vec{x} \le \vec{b}\} \subseteq \mathbb{R}^n$ be a polyhedron, let $\bar{\vec{x}} \in P$, and let

$$A^= \bar{\vec{x}} = \vec{b}^=$$

be the tight constraints for $\bar{\vec{x}}$. Then $\bar{\vec{x}}$ is extreme if and only if $\operatorname{rank}(A^=) = n$.

Lemma 13.2 Let $0 < \lambda < 1$, and let $\alpha_1, \alpha_2, \beta \in \mathbb{R}$ satisfy $\beta = \lambda\alpha_1 + (1-\lambda)\alpha_2$ together with $\alpha_1, \alpha_2 \le \beta$. Then $\alpha_1 = \alpha_2 = \beta$.

Proof. Observe that

$$\beta = \underbrace{\lambda\alpha_1}_{\le \lambda\beta} + \underbrace{(1-\lambda)\alpha_2}_{\le (1-\lambda)\beta} \le \lambda\beta + (1-\lambda)\beta = \beta.$$

Therefore equality holds throughout, so $\alpha_1 = \alpha_2 = \beta$. $\square$

Proof of Theorem 13.1. *Reverse implication.* Suppose $\operatorname{rank}(A^=) = n$ but $\bar{\vec{x}}$ is not extreme. Then $\bar{\vec{x}}$ is properly contained in a line segment between points $\vec{x}^1, \vec{x}^2 \in P$ with $\vec{x}^1 \ne \bar{\vec{x}} \ne \vec{x}^2$. Algebraically, there exists $\lambda \in (0,1)$ such that $\bar{\vec{x}} = \lambda\vec{x}^1 + (1-\lambda)\vec{x}^2$. Therefore

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

By the previous lemma, this implies

$$A^=\vec{x}^1 = \vec{b}^=$$

and
$$A^{=}\vec{x}^2 = \vec{b}^{=}.$$
Since $\operatorname{rank}(A^{=}) = n$, these equations have a unique solution, so $\vec{x}^1 = \vec{x}^2$, a contradiction.

Forward implication. We prove the contrapositive. Suppose $\operatorname{rank}(A^{=}) < n$. Then there exists $\vec{d} \neq \vec{0}$ such that $A^{=}\vec{d} = \vec{0}$. Let
$$\vec{x}^1 = \bar{\vec{x}} + \varepsilon\vec{d} \qquad \text{and} \qquad \vec{x}^2 = \bar{\vec{x}} - \varepsilon\vec{d},$$
where $\varepsilon > 0$ is sufficiently small. Then
$$\frac{1}{2}\vec{x}^1 + \frac{1}{2}\vec{x}^2 = \bar{\vec{x}},$$
so $\bar{\vec{x}}$ lies in the line segment between $\vec{x}^1$ and $\vec{x}^2$. For every tight constraint, we have
$$A^{=}\vec{x}^1 = A^{=}(\bar{\vec{x}} + \varepsilon\vec{d}) = A^{=}\bar{\vec{x}} + \varepsilon A^{=}\vec{d} = \vec{b}^{=},$$
and similarly for $\vec{x}^2$. Since the remaining constraints are strict at $\bar{\vec{x}}$, choosing ε small enough keeps both $\vec{x}^1$ and $\vec{x}^2$ feasible. Hence $\bar{\vec{x}}$ is not extreme. $\square$

Theorem 13.3 Let
$$P = \{\vec{x} \geq \vec{0} \mid A\vec{x} = \vec{b}\}$$
and assume the rows of A are independent. Then $\bar{\vec{x}}$ is extreme if and only if $\bar{\vec{x}}$ is a basic feasible solution.

Proof. Let $J = \{j \mid \bar{x}_j > 0\}$ be the support of $\bar{\vec{x}}$. We first show that $\bar{\vec{x}}$ is extreme if and only if the columns of A_J are linearly independent.

If the columns of A_J are dependent, there is a nonzero vector $\vec{d}$ supported on J such that $A\vec{d} = 0$. Because every coordinate $\bar{x}_j$ with $j \in J$ is strictly positive, for sufficiently small $\varepsilon > 0$ both $\bar{\vec{x}} + \varepsilon\vec{d}$ and $\bar{\vec{x}} - \varepsilon\vec{d}$ are nonnegative. They also satisfy $A\vec{x} = \vec{b}$, and their midpoint is $\bar{\vec{x}}$, so $\bar{\vec{x}}$ is not extreme.

Conversely, suppose $\bar{\vec{x}}$ is not extreme. Then $\bar{\vec{x}} = \lambda\vec{x}^{,1} + (1-\lambda)\vec{x}^{,2}$ for distinct feasible vectors $\vec{x}^{,1}, \vec{x}^{,2}$ and some $0 < \lambda < 1$. If $\bar{x}_j = 0$, nonnegativity forces $x_j^1 = x_j^2 = 0$. Hence the nonzero vector $\vec{d} = \vec{x}^{,1} - \vec{x}^{,2}$ is supported on J, and $A\vec{d} = 0$. Thus the columns of A_J are dependent.

If $\bar{\vec{x}}$ is a basic feasible solution for a basis B, then $J \subseteq B$, so the columns of A_J are independent and $\bar{\vec{x}}$ is extreme. Conversely, if $\bar{\vec{x}}$ is extreme, then A_J has independent columns. Since A has row rank m, we can extend J to an m-column basis B. Every coordinate outside B is zero, so $\bar{\vec{x}}$ is the basic solution associated with B; it is feasible by assumption. $\square$

Example 13.4

$$P = \{\vec{x} \geq \vec{0} \mid \begin{bmatrix} 1 & 3 & 1 & 0 \\ 2 & 2 & 0 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}\}.$$

The vector $\vec{x} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 3 \end{bmatrix}$ is a basic feasible solution. We now use the previous theorem to show that this vector is extreme.

Rewrite P as

$$P = \left\{ \vec{x} \mid \begin{bmatrix} 1 & 3 & 1 & 0 \\ 2 & 2 & 0 & 1 \\ -1 & -3 & -1 & 0 \\ -2 & -2 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \vec{x} \leq \begin{bmatrix} 1 \\ 3 \\ -1 \\ -3 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\}.$$

For $\bar{\vec{x}}$, the matrix of tight constraints is

$$A^{=} = \begin{bmatrix} 1 & 3 & 1 & 0 \\ 2 & 2 & 0 & 1 \\ -1 & -3 & -1 & 0 \\ -2 & -2 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}.$$

Rows 3 and 4 are linearly dependent on rows 1 and 2, so we may drop them. The remaining matrix has rank 4, and therefore $\bar{\vec{x}}$ is extreme.

The simplex algorithm moves from one basic feasible solution to another, and therefore from one extreme point to another.

For example, the LP (P) given by

$$\begin{aligned} \max \quad & \begin{bmatrix} 2 \\ 1 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \vec{x} \leq \begin{bmatrix} 3 \\ 2 \end{bmatrix}, \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

becomes, after adding slack variables,

$$\begin{aligned} \max \quad & \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}, \\ & \vec{x} \geq \vec{0}. \end{aligned} \tag{P'}$$

There is a natural bijection between feasible solutions of (P) and (P'):

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \longleftrightarrow \begin{bmatrix} x_1 \\ x_2 \\ 3 - x_1 - x_2 \\ 2 - x_1 \end{bmatrix}.$$

This bijection also maps extreme points of (P) to extreme points of (P'), and vice versa.

4. The Shortest Path Problem and Duality

We next develop duality, which turns arguments that establish lower bounds into a general theory with algorithmic consequences.

Shortest Path Lower Bounds

Our goal is to generate lower bounds for the shortest path problem. We begin with the cardinality case, where all edges have weight 1. The path and four pairwise disjoint st-cuts used below are shown in Figure 20.

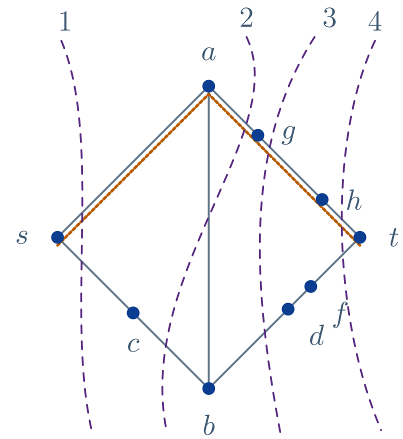


FIGURE 20

We want to show that 4 is the length of a shortest path, or equivalently a lower bound on the length of every st-path.

To this end, let Q be an arbitrary st-path. Then Q intersects the first, second, third, and fourth st-cuts. Since no two of these cuts share an edge, Q has at least four edges.

In general, let $G = (V, E)$ with $s, t \in V$ and $s \neq t$. If there exist k pairwise disjoint st-cuts, then every st-path contains at least k edges. Thus k is a lower bound on the length of every st-path.

Lecture 14 — Thursday, February 27, 2014

Weak Duality

We used an ad hoc argument to get lower bounds for the shortest path problem by assigning widths to st-cuts. Duality gives a systematic way to produce such lower bounds.

Example 14.1 Consider the linear program

$$\begin{aligned} \min \quad & \vec{c}^\top \vec{x} \\ \text{subject to} \quad & A\vec{x} \geq \vec{b}, \\ & \vec{x} \geq \vec{0}, \end{aligned}$$

where

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \\ -1 & 1 \end{bmatrix}, \qquad \vec{b} = \begin{bmatrix} 20 \\ 18 \\ 8 \end{bmatrix}, \qquad \text{and} \qquad \vec{c} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

It is easy to check that $\begin{bmatrix} 8 \\ 16 \end{bmatrix}$ and $\begin{bmatrix} 5 \\ 13 \end{bmatrix}$ are both feasible. Their objective values are 64 and 49, respectively. We want to prove that $\begin{bmatrix} 5 \\ 13 \end{bmatrix}$ is optimal by finding a lower bound of 49 on the value of every feasible solution.

Take any vector $\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \geq \vec{0}$. If we multiply the ith constraint of $A\vec{x} \geq \vec{b}$ by y_i and add them all up, we obtain

$$\vec{y}^\top A\vec{x} \geq \vec{y}^\top \vec{b}.$$

For instance, if we choose $\vec{y} = \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}$, then

$$\begin{bmatrix} 1 \\ 3 \end{bmatrix} \cdot \vec{x} \geq 44,$$

or equivalently

$$0 \geq 44 - \begin{bmatrix} 1 \\ 3 \end{bmatrix} \cdot \vec{x}.$$

Adding this to the objective function gives

$$\begin{aligned} \vec{c}^\top \vec{x} &\geq \begin{bmatrix} 2 \\ 3 \end{bmatrix} \cdot \vec{x} + 44 - \begin{bmatrix} 1 \\ 3 \end{bmatrix} \cdot \vec{x} \\ &= 44 + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \vec{x}. \end{aligned}$$

Since $\vec{x} \geq \vec{0}$, we get $\begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \vec{x} \geq 0$, and so every feasible solution satisfies $\vec{c}^\top \vec{x} \geq 44$. This is not yet enough to prove optimality, but it does show that the optimum lies between 44 and 49.

To get the best lower bound from this method, rewrite the inequality as

$$0 \geq \vec{y}^\top \vec{b} - \vec{y}^\top A\vec{x},$$

and add it to the objective:

$$\vec{c}^\top \vec{x} \geq \vec{y}^\top \vec{b} + (\vec{c}^\top - \vec{y}^\top A)\vec{x}.$$

So if we choose $\vec{y} \geq \vec{0}$ such that

$$\vec{c}^\top - \vec{y}^\top A \geq \vec{0}^\top,$$

then every feasible $\vec{x}$ satisfies

$$\vec{c}^\top \vec{x} \geq \vec{y}^\top \vec{b}.$$

Therefore the best lower bound obtainable in this way is the optimum value of

$$\begin{aligned} \max \quad & \vec{b}^\top \vec{y} \\ \text{subject to} \quad & A^\top \vec{y} \leq \vec{c}, \\ & \vec{y} \geq \vec{0}. \end{aligned}$$

This new linear program is called the **dual**. For the present example, solving the dual gives

$$\vec{y} = \begin{bmatrix} 0 \\ \frac{5}{2} \\ \frac{1}{2} \end{bmatrix} \qquad \text{and} \qquad \vec{b}^\top \vec{y} = 49.$$

Since the feasible solution $\begin{bmatrix} 5 \\ 13 \end{bmatrix}$ also has objective value 49, it is optimal.

Theorem 14.2 (Weak duality) Consider the pair of linear programs

$$\begin{aligned} \min \quad & \vec{c}^\top \vec{x} && (P) \\ \text{subject to} \quad & A\vec{x} \geq \vec{b} \\ & \vec{x} \geq \vec{0} \end{aligned}$$

$$\begin{aligned} \max \quad & \vec{b}^\top \vec{y} && (D) \\ \text{subject to} \quad & A^\top \vec{y} \leq \vec{c} \\ & \vec{y} \geq \vec{0} \end{aligned}$$

Let $\bar{\vec{x}}$ be feasible for (P) and let $\bar{\vec{y}}$ be feasible for (D). Then

$$\vec{c}^\top \bar{\vec{x}} \geq \vec{b}^\top \bar{\vec{y}}.$$

Moreover, if equality holds, then $\bar{\vec{x}}$ is optimal for (P).

Proof. Since $\bar{\vec{y}} \geq \vec{0}$ and $A\bar{\vec{x}} \geq \vec{b}$, we have

$$\vec{b}^\top \bar{\vec{y}} \leq \bar{\vec{y}}^\top (A\bar{\vec{x}}).$$

Also,

$$\bar{\vec{y}}^\top (A\bar{\vec{x}}) = (\bar{\vec{y}}^\top A)\bar{\vec{x}} = (A^\top \bar{\vec{y}})^\top \bar{\vec{x}}.$$

Since $A^\top \bar{\vec{y}} \leq \vec{c}$ and $\bar{\vec{x}} \geq \vec{0}$, it follows that

$$(A^\top \bar{\vec{y}})^\top \bar{\vec{x}} \leq \vec{c}^\top \bar{\vec{x}}.$$

Combining these inequalities gives $\vec{b}^\top \bar{\vec{y}} \leq \vec{c}^\top \bar{\vec{x}}$. If equality holds, then $\bar{\vec{x}}$ attains a value equal to a lower bound on (P), so $\bar{\vec{x}}$ is optimal. $\square$

Next time we will apply this to the shortest path formulation. The dual variables will turn out to be widths assigned to st-cuts.

Lecture 15 — Tuesday, March 04, 2014

Primal–Dual Shortest Path Methods

Example 15.1 We start with the concrete graph in Figure 21 so the primal and dual variables remain easy to interpret.

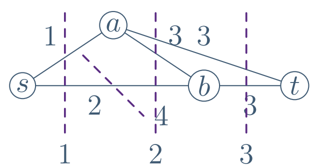


FIGURE 21

The path through a has length $d(P) = 1 + 3 = 4$. Our goal is to show that 4 is both an upper bound and a lower bound on the shortest path value, and therefore is optimal.

We begin with the LP relaxation of the shortest path cut formulation.

$$\begin{aligned} \min \quad & \begin{bmatrix} 1 \\ 2 \\ 3 \\ 3 \\ 3 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \vec{x} \geq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \qquad \text{and} \qquad \vec{x} \geq \vec{0} \end{aligned} \tag{P}$$

Remark 15.2 Recall the general weak duality pair

$$\begin{array}{rlr|rlr} \min & \vec{c}^\top \vec{x} & (P) & \max & \vec{b}^\top \vec{y} & (D) \\ \text{subject to} & A\vec{x} \geq \vec{b} & & \text{subject to} & A^\top \vec{y} \leq \vec{c} & \\ & \vec{x} \geq \vec{0} & & & \vec{y} \geq \vec{0} & \end{array}$$

Weak duality says that if $\vec{x}$ is feasible for (P) and $\vec{y}$ is feasible for (D), then $\vec{c}^\top \vec{x} \geq \vec{b}^\top \vec{y}$.

We now construct the dual of (P) and look for a feasible vector $\bar{\vec{y}}$ for that dual. By weak duality, the value of $\bar{\vec{y}}$ is a lower bound on the optimum value of (P), which in turn is a lower bound on the length of a shortest path.

For the current problem, dualizing gives

$$\begin{array}{rl} \max & \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \cdot \vec{y} \\ \text{subject to} & \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \vec{y} \leq \begin{bmatrix} 1 \\ 2 \\ 3 \\ 3 \\ 3 \end{bmatrix} \qquad \text{and} \qquad \vec{y} \geq \vec{0} \end{array} \tag{D}$$

The dual variables assign nonnegative weights to st-cuts. There is one dual constraint for each edge. For example, the constraint for edge sa is

$$y_{\{s\}} + y_{\{s,b\}} \leq 1 = c_{sa},$$

because those are exactly the st-cuts that use edge sa. Similarly, the constraint for edge bt is

$$y_{\{s,b\}} + y_{\{s,a,b\}} \leq 3 = c_{bt}.$$

So the dual problem is trying to maximize the total width assigned to st-cuts. This dual can be read geometrically as assigning nonnegative widths to cuts, subject to edge capacity limits.

In general, let $G = (V, E)$ with $s, t \in V$, $s \neq t$, and $c_e \geq 0$ for all $e \in E$. The shortest path problem can be written as

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e x_e \\ \text{subject to} \quad & \sum_{e \in \delta(U)} x_e \geq 1, \quad \text{for every } st\text{-cut } \delta(U), \\ & \vec{x} \geq \vec{0}, \\ & \vec{x} \text{ integer}. \end{aligned}$$

Dropping the integrality constraint gives the LP relaxation

$$\begin{aligned} \min \quad & \vec{c}^{\top} \vec{x} \\ \text{subject to} \quad & A\vec{x} \geq \vec{1}, \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

The matrix A is the cut–edge incidence matrix.

$$A = \begin{array}{cl} \left[\begin{array}{ccc} & \vdots & \\ \hline \dots & 1 & \dots \\ & \vdots & \end{array}\right] & \leftarrow u \end{array}$$

Its entry in row U and column e is 1 if $e \in \delta(U)$ and 0 otherwise.

The dual of this LP relaxation is

$$\begin{aligned} \max \quad & \vec{1}^{\top} \vec{y} \\ \text{subject to} \quad & A^{\top} \vec{y} \leq \vec{c}, \\ & \vec{y} \geq \vec{0}. \end{aligned}$$

Equivalently, we may write it as

$$\begin{aligned} \max \quad & \sum_{\delta(U) \text{ is an } st\text{-cut}} y_U \\ \text{subject to} \quad & \sum_{U : e \in \delta(U)} y_U \leq c_e, \quad \text{for every } e \in E, \\ & \vec{y} \geq \vec{0}, \end{aligned}$$

where the sum is over all st-cuts $\delta(U)$.

This interpretation suggests a shortest path algorithm based on growing cut widths.

Lecture 16 — Thursday, March 06, 2014

We now combine primal and dual viewpoints to obtain a constructive shortest path method.

A small worked instance makes the primal–dual mechanism transparent before the general method; its graph and two highlighted cuts appear in Figure 22.

Example 16.1

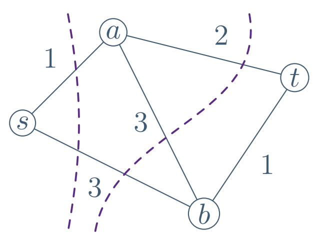


FIGURE 22

For this graph, the shortest path formulation and its dual are given below.

$$
\begin{array}{ll}
\min & \begin{bmatrix} 1 \\ 3 \\ 3 \\ 2 \\ 1 \end{bmatrix} \cdot \vec{x} \\
\text{subject to} & \begin{array}{r|ccccc}
 & sa & sb & ab & at & bt \\ \hline
\{s\} & 1 & 1 & & & \\
\{s,a\} & & 1 & 1 & 1 & \\
\{s,b\} & 1 & & 1 & & 1 \\
\{s,a,b\} & & & & 1 & 1
\end{array} \; \vec{x} \geq \vec{1}
\end{array} \qquad (P)
$$

$$
\begin{array}{ll}
\max & \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \cdot \vec{y} \\
\text{subject to} & \begin{array}{c|cccc}
 & \{s\} & \{s,a\} & \{s,b\} & \{s,a,b\} \\
\hline
sa & 1 & & 1 & \\
sb & 1 & 1 & & \\
ab & & 1 & 1 & \\
at & & 1 & & 1 \\
bt & & & 1 & 1
\end{array} \vec{y} \leq \begin{bmatrix} 1 \\ 3 \\ 3 \\ 2 \\ 1 \end{bmatrix}
\end{array} \qquad (D)
$$

We claim that the path $P = sa, at$ is shortest. On the dual side, take the feasible vector $\vec{y}$ with $y_{\{s\}} = 1$, $y_{\{s,a\}} = 2$, and all other components equal to 0. Its objective value is $1+2=3$. On the primal side, take the feasible vector $\vec{x}$ with $x_{sa} = 1$, $x_{at} = 1$, and all other components equal to 0. Its objective value is also 3. By weak duality, both solutions are optimal.

Definition 16.2 For a fixed dual vector $\vec{y}$, define the **slack** of an edge e by

$$\text{slack}_y(e) = c_e - (\text{total width of the } st\text{-cuts using } e).$$

Definition 16.3 A **directed *st*-path** is a path of the form $s \to \cdots \to t$.

We now state the algorithm that implements the primal–dual idea step by step.

Algorithm 2: Primal–Dual Shortest Path Algorithm

Input: A graph $G = (V, E)$ containing an st-path, with edge lengths $c_e \geq 0$ and distinct vertices $s, t \in V$

Output: A directed shortest st-path together with a dual feasible cut width vector

Initialize $y \leftarrow 0$ and $U \leftarrow \{s\}$;
while $t \notin U$ **do**
  choose an edge $ab \in \delta(U)$ of minimum slack, labelled so that $a \in U$ and $b \notin U$;
  increase y_U by $\text{slack}_y(ab)$;
  enlarge U to $U \cup \{b\}$;
  orient the chosen edge as an arc $a \to b$ and record a as the predecessor of b;
end
return *the unique directed st-path in the recorded predecessor tree*;

Example 16.4

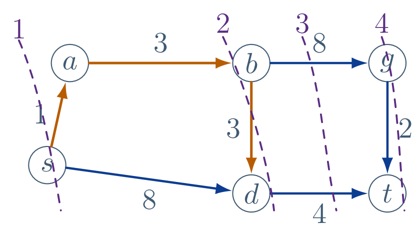


Figure 23

For the instance in Figure 23, we begin with $y = 0$ and $U = \{s\}$. Then $\text{slack}(sa) = 1$ and $\text{slack}(sd) = 8$, so we set $y_{\{s\}} = 1$ and enlarge U to $\{s, a\}$. Next we have $\text{slack}(ab) = 3$ and $\text{slack}(sd) = 7$, so we set $y_{\{s,a\}} = 3$ and enlarge U to $\{s, a, b\}$. At the next step, $\text{slack}(bg) = 8$, $\text{slack}(bd) = 3$, and $\text{slack}(sd) = 8 - 1 - 3 = 4$, so we set $y_{\{s,a,b\}} = 3$ and enlarge U again. Continuing in this way produces the directed path sa, ab, bd, dt, whose total length is $1 + 3 + 3 + 4 = 11$.

Does the algorithm terminate? Most assuredly, because at every iteration one new vertex is added. Does it work? It does. Does it give an optimal solution? That is not immediately clear. To prove that it does, we restate the relevant optimality condition.

Definition 16.5 e is an **equality edge** if it has slack 0, that is, if it corresponds to a tight constraint.

Definition 16.6 An st-cut $\delta(U)$ is **active** if it has positive width.

Theorem 16.7 (Optimality Criterion) Let y be feasible for the dual problem (D), and let Q be an st-path. If

1. The constraints of (D) corresponding to the edges of Q are tight, and
2. Q uses exactly one edge from each active cut,

then Q is a shortest st-path.

Proof. The total st-cut width is

$$\sum_{U:y_U>0} y_U = \sum_{e\in Q}\left(\sum_{\substack{U:y_U>0\\ e\in\delta(U)}} y_U\right) = \sum_{e\in Q} c_e,$$

where the first equality is due to Assumption (2), and the second is due to Assumption (1). Hence Q is optimal. ☐

Lemma 16.8 Suppose that the primal–dual shortest path algorithm has terminated and the following properties hold:

1. $\bar{y}$ is feasible for the dual problem (D).
2. All chosen arcs are equality arcs (i.e., they have slack zero).
3. For every cut W with positive width, the arc configuration in Figure 24 is not possible:

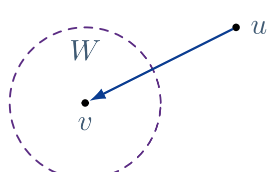


FIGURE 24

4. For all vertices $v \in U$ with $v \neq s$, there exists a directed path from s to v.

Then the output of the algorithm is optimal.

Proof. The algorithm terminates once $t \in U$. (4) says that t is reachable from s; call that st-path Q. (2) then says that for all arcs in Q, the corresponding dual constraint is tight.

Now, because $\bar{y}$ is feasible for (D) by (1), Theorem 16.7 shows that it suffices that every cut with positive width is crossed exactly once by Q. Every active cut W arose as a set U before t entered it, so $s \in W$ and $t \notin W$; consequently Q must leave W at least once. Condition (3) forbids any chosen arc from entering W. Once Q leaves W, it therefore cannot return, so it crosses W exactly once. □

To prove that the primal–dual shortest path algorithm always produces an optimal st-path, it suffices to prove that the conditions in Lemma 16.8 hold at every step of the algorithm; that is, they are **invariants**.

Lemma 16.9 (1)–(4) in Lemma 16.8 are maintained at every step of the primal–dual shortest path algorithm.

Proof. We prove this by induction on the iteration number.

Initially, $y = 0$ and $U = \{s\}$. Then (1) holds because all dual constraints have nonnegative slack, (2) is vacuous (no chosen arcs yet), (3) is vacuous (no active cut), and (4) is immediate. This proves the base case.

For the induction step, assume (1)–(4) hold at the beginning of an iteration, with current pair (U, y). Let $ab \in \delta(U)$ be an edge of minimum slack, labelled with $a \in U$ and $b \notin U$. Put $\Delta = \text{slack}_y(ab)$. The update sets

$$y'_U = y_U + \Delta \qquad \text{and} \qquad y'_W = y_W \text{ for } W \neq U,$$

and replaces U by $U' = U \cup \{b\}$, adding the arc $a \to b$.

We verify each assumption for (U', y').

(1): By the inductive hypothesis, every edge has nonnegative slack under y. For any edge $f \in \delta(U)$,

$$\text{slack}_{y'}(f) = \text{slack}_y(f) - \Delta = \text{slack}_y(f) - \text{slack}_y(ab) \geq 0,$$

because ab has minimum slack in $\delta(U)$. For edges $f \notin \delta(U)$, the value of $\sum_{W:\, f \in \delta(W)} y_W$ is unchanged, so their slack is unchanged and remains nonnegative.

(2): For the newly added arc ab,

$$\text{slack}_{y'}(ab) = \text{slack}_y(ab) - \Delta = 0.$$

Any previously chosen arc had both endpoints in U at the start of this iteration (once chosen, both endpoints stay in all later supersets of U), hence it is not in $\delta(U)$ and its slack is unchanged. By induction, those slacks were already zero.

(3): Consider a cut W with $y'_W > 0$. If $W \neq U$, then $y_W > 0$ already before the update, so the property for previously chosen arcs follows by induction; and the new arc $a \to b$ cannot enter W because such old cuts are subsets of U, while $b \notin U$. If $W = U$ (newly active), then previously chosen arcs have both endpoints in U, so they do not cross $\delta(U)$, and the new arc $a \to b$ leaves U (it does not enter).

(4) reachability in U': Vertices already in U remain reachable by induction. Since $a \in U$, there is a directed path from s to a; appending the new arc $a \to b$ gives a directed path from s to b. Hence every vertex in U' is reachable from s.

Therefore (1)–(4) are preserved at every iteration. $\square$

Combining Lemmas 16.8 and 16.9 proves the following theorem:

Theorem 16.10 The primal–dual shortest path algorithm always terminates with a feasible solution to (D). Moreover, the dual solution gives rise to an optimal st-path.

Lecture 17 — Tuesday, March 11, 2014

Integral Optimal Solutions for Shortest Paths

Returning to the cut formulation of the shortest path problem, consider our formulation:

$$\begin{aligned} \min \sum_{e \in E} c_e x_e & \\ \text{subject to} \sum_{e \in \delta(U)} x_e \geq 1 & \quad \text{for every } st\text{-cut } \delta(U), \\ x_e \geq 0 & \quad \text{for every } e \in E. \end{aligned}$$

Proposition 17.1 If G contains an st-path and $c_e \geq 0$ for every $e \in E$, then this linear program has an optimal solution that is integral.

Proof. Let P be a shortest st-path, and define

$$x_e = \begin{cases} 1 & \text{if } e \in P, \\ 0 & \text{otherwise.} \end{cases}$$

Then x is feasible for the primal problem. Moreover, the vector $\vec{y}$ produced by the shortest path algorithm is feasible for the dual problem, and the objective value of x is equal to the objective value of $\vec{y}$. Weak duality therefore implies that both solutions are optimal. $\square$

This does not mean that every feasible linear program has an integer optimal solution.

Example 17.2 Consider

$$\begin{aligned} \min\ & x_1 + x_2 + x_3 \\ \text{subject to } & x_1 + x_2 \geq 1, \\ & x_1 + x_3 \geq 1, \\ & x_2 + x_3 \geq 1, \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

The corresponding integer program has optimal value 2, whereas the linear relaxation has optimal value 3/2.

Duality Rules Beyond Standard Form

We have already seen the standard dual pair

$$\begin{aligned} \min \quad & \vec{c}^\top \vec{x} & (P_{\min}) \\ \text{subject to} \quad & A\vec{x} \geq \vec{b}, \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

$$\begin{aligned} \max \quad & \vec{b}^\top \vec{y} & (P_{\max}) \\ \text{subject to} \quad & A^\top \vec{y} \leq \vec{c}, \\ & \vec{y} \geq \vec{0}. \end{aligned}$$

Weak duality says that if $\vec{x}$ is feasible for $(P_{\min})$ and $\vec{y}$ is feasible for $(P_{\max})$, then

$$\vec{c}^\top \vec{x} \geq \vec{b}^\top \vec{y}.$$

The same principle continues to hold for more general sign conventions. The following table summarizes the full picture:

$P_{\max}$	$\Longleftrightarrow$	$P_{\min}$
$\max \vec{c}^\top \vec{x}$ subject to $\begin{array}{c} A\vec{x} \ ?\ \vec{b} \\ \vec{x} \ ? \end{array}$	DUALITY	$\min \vec{b}^\top \vec{y}$ subject to $\begin{array}{c} A^\top\vec{y} \ ?\ \vec{c} \\ \vec{y} \ ? \end{array}$
$x_j \geq 0, x_k \leq 0, x_l$ free	$\updownarrow$	constraint j is "$\geq$", k is "$\leq$", l is "$=$"
constraint j is "$=$", k is "$\geq$", l is "$\leq$"	$\updownarrow$	y_j free, $y_k \leq 0, y_l \geq 0$

For example, consider

$$\begin{aligned} \max \quad & \begin{bmatrix} 4 \\ -1 \\ 3 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 3 & -1 & 2 \\ 2 & 1 & 1 \end{bmatrix} \vec{x} \leq \begin{bmatrix} 4 \\ 5 \end{bmatrix}, \\ & x_1 \geq 0,\ x_2 \leq 0,\ x_3 \text{ free.} \end{aligned}$$

Its dual is

$$\begin{aligned} \min \quad & \begin{bmatrix} 4 \\ 5 \end{bmatrix} \cdot \vec{y} \\ \text{subject to} \quad & \begin{bmatrix} 3 & 2 \\ -1 & 1 \\ 2 & 1 \end{bmatrix} \vec{y} \begin{array}{c} \geq \\ \leq \\ = \end{array} \begin{bmatrix} 4 \\ -1 \\ 3 \end{bmatrix}, \\ & \vec{y} \geq \vec{0}. \end{aligned}$$

Similarly, the minimization problem

$$\begin{aligned} \min \quad & \begin{bmatrix} 1 \\ 2 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 3 & 4 \\ 5 & 6 \end{bmatrix} \vec{x} = \begin{bmatrix} 7 \\ 8 \end{bmatrix}, \\ & \vec{x} \geq \vec{0} \end{aligned}$$

has dual

$$\begin{aligned} \max \quad & \begin{bmatrix} 7 \\ 8 \end{bmatrix} \cdot \vec{y} \\ \text{subject to} \quad & \begin{bmatrix} 3 & 5 \\ 4 & 6 \end{bmatrix} \vec{y} \leq \begin{bmatrix} 1 \\ 2 \end{bmatrix} \\ & \vec{y} \text{ free.} \end{aligned}$$

Applying the duality rules twice returns the original problem. It is therefore always useful to know how to identify the dual of a nonstandard formulation.

Weak duality also remains valid in these more general settings. For example, consider the pair

$$\begin{array}{rll} \max & \vec{c}^\top \vec{x} & (P_{\max}) \\ \text{subject to} & A\vec{x} \leq \vec{b}, & \\ & \vec{x} \text{ free.} & \end{array}$$

$$\begin{array}{rll} \min & \vec{b}^\top \vec{y} & (P_{\min}) \\ \text{subject to} & A^\top \vec{y} = \vec{c}, & \\ & \vec{y} \geq \vec{0}. & \end{array}$$

Proposition 17.3 If $\vec{x}$ is feasible for $(P_{\max})$ and $\vec{y}$ is feasible for $(P_{\min})$, then $\vec{c}^\top \vec{x} \leq \vec{b}^\top \vec{y}$.

Proof. Since $\vec{x}$ is feasible for $(P_{\max})$, we can write $A\vec{x} + \vec{s} = \vec{b}$ with $\vec{s} \geq \vec{0}$. Since $\vec{y}$ is feasible for $(P_{\min})$, we have $A^\top \vec{y} = \vec{c}$ and $\vec{y} \geq \vec{0}$. Therefore

$$\vec{c}^\top \vec{x} = (\vec{y}^\top A)\vec{x} = \vec{y}^\top (A\vec{x}) = \vec{y}^\top (\vec{b} - \vec{s}) = \vec{y}^\top \vec{b} - \vec{y}^\top \vec{s} \leq \vec{y}^\top \vec{b} = \vec{b}^\top \vec{y}.$$

□

Weak duality has several immediate consequences:

1. If $\vec{x}$ is feasible for $(P_{\max})$ and $\vec{y}$ is feasible for $(P_{\min})$ with $\vec{c}^\top \vec{x} = \vec{b}^\top \vec{y}$, then both solutions are optimal.
2. If $(P_{\max})$ is unbounded, then $(P_{\min})$ is infeasible.
3. If $(P_{\min})$ is unbounded, then $(P_{\max})$ is infeasible.
4. If both $(P_{\max})$ and $(P_{\min})$ are feasible, then both have optimal solutions.

The final statement follows from the fundamental theorem of linear programming. Once feasibility is known for both problems, the previous two consequences rule out the possibility that either problem is unbounded.

Lecture 18 — Thursday, March 13, 2014

This topic strengthens weak duality ideas and introduces conditions that characterize optimal pairs precisely.

Strong Duality

Strong duality says more than weak duality. If one member of a dual pair has an optimal solution, then the other member also has an optimal solution, and the two optimal objective values are equal. If neither side has an optimal solution, there are three possibilities: the primal is unbounded and the dual infeasible, the primal is infeasible and the dual unbounded, or both problems are infeasible.

Theorem 18.1 (Strong Duality for SEF) Consider the dual pair

$$\begin{aligned} \max \quad & \vec{c}^{\top}\vec{x} & \qquad (P_{\max}) \\ \text{subject to} \quad & A\vec{x} = \vec{b}, \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

$$\begin{aligned} \min \quad & \vec{b}^{\top}\vec{y} & \qquad (P_{\min}) \\ \text{subject to} \quad & A^{\top}\vec{y} \geq \vec{c}, \\ & \vec{y} \text{ free}. \end{aligned}$$

If $(P_{\max})$ has an optimal solution, then $(P_{\min})$ has an optimal solution as well, and the optimal objective values are equal.

Proof. Since $(P_{\max})$ has an optimal solution, the simplex algorithm terminates at an optimal basis B. Rewrite the objective in canonical form with respect to B:

$$\begin{aligned} \max \quad & (\vec{c}^{\top} - \vec{y}^{\top}A)\vec{x} + \vec{b}^{\top}\vec{y} \\ \text{subject to} \quad & A_B^{-1}A\vec{x} = A_B^{-1}\vec{b}, \\ & \vec{x} \geq \vec{0}, \end{aligned}$$

where $\vec{y}^{\top} = \vec{c}_B^{\top}A_B^{-1}$. Let $\bar{\vec{x}}$ be the basic feasible solution associated with B. Because the simplex algorithm has terminated, all reduced costs are nonpositive, so

$$\vec{c}^\top - \vec{y}^\top A \le \vec{0}^\top.$$

Equivalently, $A^\top \vec{y} \ge \vec{c}$, so $\vec{y}$ is feasible for $(P_{\min})$. For the basis B, the constant term in the canonical objective is $\vec{b}^\top \vec{y}$. Rewriting the problem into canonical form does not change the objective value of any feasible solution, so the basic feasible solution $\bar{\vec{x}}$ satisfies

$$\vec{c}^\top \bar{\vec{x}} = \vec{b}^\top \vec{y}.$$

Weak duality now implies that $\bar{\vec{x}}$ and $\vec{y}$ are optimal. □

For the SEF dual pair above, it is enough to verify the following three properties in order to prove that $\vec{x}$ and $\vec{y}$ are optimal:

(A) $\vec{x}$ is feasible for $(P_{\max})$.

(B) $\vec{y}$ is feasible for $(P_{\min})$.

(C) $\vec{c}^\top \vec{x} = \vec{b}^\top \vec{y}$.

The simplex algorithm maintains (A) and (C) while working toward (B). There are other options; for example, the **dual simplex algorithm** maintains (B) and (C) while working toward (A), while the **interior point algorithm** maintains primal and dual feasibility and drives the gap in (C) to zero. We do not cover these latter two algorithms in this course.

Complementary Slackness

Theorem 18.2 (Complementary slackness) Consider the dual pair

$$\begin{aligned} \max \quad & \vec{c}^\top \vec{x} & \qquad (P_{\max}) \\ \text{subject to} \quad & A\vec{x} \le \vec{b}, & \\ & \vec{x} \text{ free.} & \end{aligned}$$

$$\begin{aligned} \min \quad & \vec{b}^\top \vec{y} & \qquad (P_{\min}) \\ \text{subject to} \quad & A^\top \vec{y} = \vec{c}, & \\ & \vec{y} \ge \vec{0}. & \end{aligned}$$

Let $\bar{\vec{x}}$ be feasible for $(P_{\max})$, and let $\bar{\vec{y}}$ be feasible for $(P_{\min})$. Write

$$\bar{\vec{s}} = \vec{b} - A\bar{\vec{x}}.$$

Then $\bar{\vec{x}}$ and $\bar{\vec{y}}$ are optimal if and only if the **complementary slackness conditions**

$$\bar{y}_i \bar{s}_i = 0 \quad \text{for every } i = 1, \ldots, m$$

hold. Equivalently, for each i, either $\bar{y}_i = 0$ or the ith primal constraint is tight.

Proof. Write the primal constraints as $A\vec{x} + \vec{s} = \vec{b}$ with $\vec{s} \geq \vec{0}$. If $\vec{x}$ is feasible for $(P_{\max})$ and $\vec{y}$ is feasible for $(P_{\min})$, then

$$\vec{c}^\top \vec{x} = \vec{y}^\top A\vec{x} = \vec{y}^\top (\vec{b} - \vec{s}) = \vec{y}^\top \vec{b} - \vec{y}^\top \vec{s} \leq \vec{b}^\top \vec{y}.$$

By strong duality, $\vec{x}$ and $\vec{y}$ are both optimal exactly when equality holds. Since $\vec{y} \geq \vec{0}$ and $\vec{s} \geq \vec{0}$, equality holds if and only if

$$\vec{y}^\top \vec{s} = \sum_{i=1}^{m} y_i s_i = 0,$$

which is equivalent to $y_i s_i = 0$ for every $i = 1, \ldots, m$. Thus, for each primal constraint, either the corresponding dual variable is zero or the primal constraint is tight. $\square$

Example 18.3 Consider the pair

$$\begin{aligned} \max \quad & \begin{bmatrix} 5 \\ 3 \\ 5 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 1 & 2 & -1 \\ 3 & 1 & 2 \\ -1 & 1 & 1 \end{bmatrix} \vec{x} \leq \begin{bmatrix} 2 \\ 4 \\ -1 \end{bmatrix}. \end{aligned}$$

and

$$\begin{aligned} \min \quad & \begin{bmatrix} 2 \\ 4 \\ -1 \end{bmatrix} \cdot \vec{y} \\ \text{subject to} \quad & \begin{bmatrix} 1 & 3 & -1 \\ 2 & 1 & 1 \\ -1 & 2 & 1 \end{bmatrix} \vec{y} = \begin{bmatrix} 5 \\ 3 \\ 5 \end{bmatrix}, \\ & \vec{y} \geq \vec{0}. \end{aligned}$$

The vectors

$$\vec{x} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \qquad \text{and} \qquad \vec{y} = \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}$$

are both feasible, and their objective values agree:

$$\begin{bmatrix} 5 \\ 3 \\ 5 \end{bmatrix} \cdot \vec{x} = 7 = \begin{bmatrix} 2 \\ 4 \\ -1 \end{bmatrix} \cdot \vec{y}.$$

Hence they are optimal. The complementary slackness conditions are

$$\begin{cases} y_1 = 0 \text{ or } x_1 + 2x_2 - x_3 = 2, \\ y_2 = 0 \text{ or } 3x_1 + x_2 + 2x_3 = 4, \\ y_3 = 0 \text{ or } -x_1 + x_2 + x_3 = -1. \end{cases}$$

Lecture 19 — Tuesday, March 18, 2014

We begin with one more example illustrating complementary slackness.

Example 19.1 Consider the primal–dual pair

$$\begin{aligned} \max \quad & \begin{bmatrix} 6 \\ 2 \\ 4 \end{bmatrix} \cdot \vec{x} & \qquad (P) \\ \text{subject to} \quad & \begin{bmatrix} 1 & 2 & -1 \\ 3 & 1 & 2 \end{bmatrix} \vec{x} \leq \begin{bmatrix} 2 \\ 4 \end{bmatrix}. \end{aligned}$$

and

$$\min \quad \begin{bmatrix} 2 \\ 4 \end{bmatrix} \cdot \vec{y} \qquad (D)$$

$$\text{subject to} \quad \begin{bmatrix} 1 & 3 \\ 2 & 1 \\ -1 & 2 \end{bmatrix} \vec{y} = \begin{bmatrix} 6 \\ 2 \\ 4 \end{bmatrix},$$

$$\vec{y} \geq \vec{0}.$$

The complementary slackness conditions are

$$\begin{cases} y_1 = 0 \text{ or } x_1 + 2x_2 - x_3 = 2, \\ y_2 = 0 \text{ or } 3x_1 + x_2 + 2x_3 = 4. \end{cases}$$

The vectors $\vec{x} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$ and $\vec{y} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$ are feasible for (P) and (D), respectively. They satisfy the complementary slackness conditions, so both are optimal.

Cone of Tight Constraints

The next step is to convert complementary slackness intuition into a geometric condition.

Definition 19.2 Let $a^1, \ldots, a^k \in \mathbb{R}^n$. The set

$$\operatorname{cone}(a^1, \ldots, a^k) := \left\{ \sum_{i=1}^{k} \lambda_i a^i \mid \lambda_i \geq 0 \text{ for } i = 1, \ldots, k \right\}$$

is called the **cone generated by $\boldsymbol{a^1}, \ldots, \boldsymbol{a^k}$**.

Definition 19.3 Let $P = \{\vec{x} \mid A\vec{x} \leq \vec{b}\}$ be a polyhedron, and let $\bar{\vec{x}} \in P$. Define

$$J(\bar{\vec{x}}) = \{i \mid \text{row}_i(A)\bar{\vec{x}} = b_i\}.$$

Then

$$\operatorname{cone}(\text{row}_i(A)^\top \mid i \in J(\bar{\vec{x}}))$$

is called the **cone of tight constraints for $\bar{\vec{x}}$**.

Example 19.4 Consider

$$\max \quad \begin{bmatrix} 3/2 \\ 1/2 \end{bmatrix} \cdot \vec{x} \qquad (P)$$
$$\text{subject to} \quad \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \vec{x} \leq \begin{bmatrix} 3 \\ 2 \\ 2 \\ 0 \\ 0 \end{bmatrix}.$$

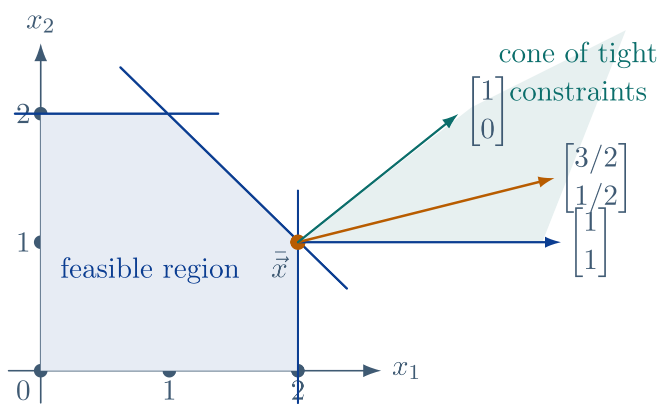


FIGURE 25

In Figure 25, the cone of tight constraints at $\bar{\vec{x}} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ is

$$\text{cone}\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right).$$

The point $\bar{\vec{x}}$ is optimal because

$$\begin{bmatrix} 3/2 \\ 1/2 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Thus the objective vector $\vec{c}$ lies in the cone generated by the normals of the tight constraints at $\bar{\vec{x}}$.

The dual of (P) is

$$\begin{aligned} \min \quad & \begin{bmatrix} 3 \\ 2 \\ 2 \\ 0 \\ 0 \end{bmatrix} \cdot \vec{y} & (D) \\ \text{subject to} \quad & \begin{bmatrix} 1 & 1 & 0 & -1 & 0 \\ 1 & 0 & 1 & 0 & -1 \end{bmatrix} \vec{y} = \begin{bmatrix} 3/2 \\ 1/2 \end{bmatrix}, \\ & \vec{y} \geq \vec{0}. \end{aligned}$$

Taking $\vec{y} = \begin{bmatrix} 1/2 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ gives a feasible dual solution. The complementary slackness conditions are satisfied with the primal point $\bar{\vec{x}}$, so both solutions are optimal.

Conversely, if $\bar{\vec{x}}$ is known to be optimal, then complementary slackness forces the multipliers of all nontight constraints to be zero. The same calculation then shows that $\vec{c}$ lies in the cone generated by the tight constraints at $\bar{\vec{x}}$.

Theorem 19.5 (Cone characterization of optimality) Let $\bar{\vec{x}}$ be feasible for

$$\begin{aligned} \max \quad & \vec{c}^\top \vec{x} & (P) \\ \text{subject to} \quad & A\vec{x} \leq \vec{b}. \end{aligned}$$

Then $\bar{\vec{x}}$ is optimal if and only if $\vec{c}$ lies in the cone of tight constraints for $\bar{\vec{x}}$.

Proof. We prove both directions using complementary slackness for the pair

$$\begin{aligned} \max \quad & \vec{c}^\top \vec{x} & (P) \\ \text{subject to} \quad & A\vec{x} \leq \vec{b}, \end{aligned} \qquad \begin{aligned} \min \quad & \vec{b}^\top \vec{y} & (D) \\ \text{subject to} \quad & A^\top \vec{y} = \vec{c}, \\ & \vec{y} \geq \vec{0}. \end{aligned}$$

Reverse implication. Assume $\vec{c}$ lies in the cone of tight constraints for $\bar{\vec{x}}$. Then

$$\vec{c} = \sum_{i \in J(\bar{\vec{x}})} \lambda_i \operatorname{row}_i(A)^\top \quad \text{with } \lambda_i \geq 0.$$

Define $\bar{\vec{y}}$ by $\bar{y}_i = \lambda_i$ for $i \in J(\bar{\vec{x}})$ and $\bar{y}_i = 0$ otherwise. Then $\bar{\vec{y}} \geq 0$, and

$$A^\top \bar{\vec{y}} = \sum_{i=1}^{m} \bar{y}_i \operatorname{row}_i(A)^\top = \vec{c},$$

so $\bar{\vec{y}}$ is feasible for (D). Now for every constraint i of (P), either $i \in J(\bar{\vec{x}})$ (so constraint i is tight at $\bar{\vec{x}}$) or $i \notin J(\bar{\vec{x}})$ (so $\bar{y}_i = 0$). Hence complementary slackness holds for $(\bar{\vec{x}}, \bar{\vec{y}})$, and therefore $\bar{\vec{x}}$ is optimal.

Forward implication. Assume $\bar{\vec{x}}$ is optimal for (P). By strong duality, there exists an optimal dual feasible vector $\bar{\vec{y}}$ for (D). Complementary slackness implies: for every i, if constraint i is not tight at $\bar{\vec{x}}$, then $\bar{y}_i = 0$. Using $A^\top \bar{\vec{y}} = \vec{c}$,

$$\vec{c} = \sum_{i=1}^{m} \bar{y}_i \operatorname{row}_i(A)^\top,$$

where nonzero coefficients occur only for $i \in J(\bar{\vec{x}})$, and all coefficients are nonnegative because $\bar{\vec{y}} \geq 0$. Therefore $\vec{c}$ is in the cone generated by $\{\operatorname{row}_i(A)^\top : i \in J(\bar{\vec{x}})\}$, i.e., in the cone of tight constraints for $\bar{\vec{x}}$. $\square$

Historical Note (2026) The subject of linear programming was developed by Leonid Kantorovich in the late 1930s, and George Dantzig published the simplex algorithm in 1947. Dantzig later recalled that John von Neumann sketched the theory of duality within minutes of hearing Dantzig explain linear programming.

Lecture 20 — Thursday, March 20, 2014

5. Solving Integer Programs

Convex Hull Formulation of Integer Programs

We now transition to integrality restrictions, where geometry and duality still guide modelling but the computational landscape changes. A simple integer programming example already suggests the main idea.

Example 20.1 Consider the integer program

$$\begin{aligned} \max \quad & x_1 \\ \text{subject to} \quad & x_1 + x_2 \leq 5/2, \\ & x_1, x_2 \geq 0, \\ & x_1, x_2 \in \mathbb{Z}. \end{aligned}$$

Because $x_1 + x_2$ is integral on every feasible integer point, this is equivalent to

$$\begin{aligned} \max \quad & x_1 \\ \text{subject to} \quad & x_1 + x_2 \leq 2, \\ & x_1, x_2 \geq 0, \\ & x_1, x_2 \in \mathbb{Z}. \end{aligned}$$

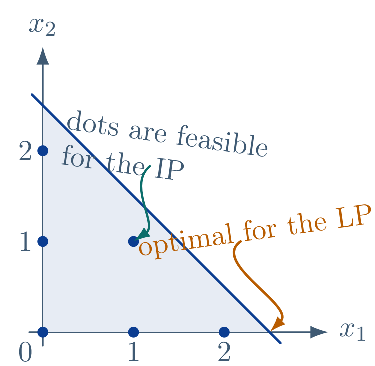


FIGURE 26

As Figure 26 suggests, we can sometimes solve an integer program by replacing its feasible integer points with their convex hull.

Definition 20.2 Let $S \subseteq \mathbb{R}^n$. The **convex hull** of S, denoted $\operatorname{conv}(S)$, is the smallest convex set containing S.

That is, the convex hull of S is the intersection of all convex sets that contain S.

Theorem 20.3 (Fundamental theorem of integer programming) Let $P = \{x \mid Ax \leq b\}$ and suppose that all entries of A and b are rational. Then

$$\operatorname{conv}\{x \in P \mid x \in \mathbb{Z}^n\}$$

is a polyhedron.

The proof is long and is omitted.

Now consider the integer program

$$\begin{aligned} \max \quad & \vec{c}^{\top}\vec{x} \\ \text{subject to} \quad & A\vec{x} \leq \vec{b}, \\ & \vec{x} \geq \vec{0}, \\ & \vec{x} \in \mathbb{Z}^n. \end{aligned}$$

By Theorem 20.3, the convex hull of its feasible integer points is a polyhedron of the form

$$\{\vec{x} \mid A'\vec{x} \leq \vec{b}',\ \vec{x} \geq \vec{0}\}.$$

If we could compute A' and $\vec{b}'$, then solving the corresponding linear program would solve the original integer program. The difficulty is that this convex hull description is usually impractical to compute directly.

Cutting Plane Methods

Instead of describing the entire convex hull, we approximate it by gradually adding valid inequalities.

Example 20.4 Consider

$$\begin{aligned} \max \quad & \begin{bmatrix} 2 \\ 5 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 1 & 4 \\ 1 & 1 \end{bmatrix} \vec{x} \leq \begin{bmatrix} 8 \\ 4 \end{bmatrix}, \\ & \vec{x} \geq \vec{0}, \\ & \vec{x} \in \mathbb{Z}^2. \end{aligned}$$

If we drop the integrality restriction, the linear relaxation has optimal solution $\bar{\vec{x}} = \begin{bmatrix} 8/3 \\ 4/3 \end{bmatrix}$. We therefore look for a linear inequality that is satisfied by every feasible integer solution but violated by $\bar{\vec{x}}$. Such an inequality is called a **cutting plane**.

For example, the inequality $x_1 + 3x_2 \leq 6$ is valid for the integer program. To see this, the first constraint forces the nonnegative integer x_2 to lie in $\{0, 1, 2\}$. If $x_2 = 0$, the second constraint gives $x_1 \leq 4$; if $x_2 = 1$, it gives $x_1 \leq 3$; and if $x_2 = 2$, the first constraint gives $x_1 = 0$. In every case, $x_1 + 3x_2 \leq 6$.

Adding this inequality to the relaxation gives

$$\begin{aligned} \max \quad & \begin{bmatrix} 2 \\ 5 \end{bmatrix} \cdot \vec{x} \qquad\qquad (P') \\ \text{subject to} \quad & \begin{bmatrix} 1 & 4 \\ 1 & 1 \\ 1 & 3 \end{bmatrix} \vec{x} \leq \begin{bmatrix} 8 \\ 4 \\ 6 \end{bmatrix}, \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

The point $\begin{bmatrix} 3 \\ 1 \end{bmatrix}$ is optimal for (P'), with objective value 11, and is integral. It is therefore also optimal for the original integer program.

This leads to the following general cutting plane procedure.

Algorithm 3: Conceptual Cutting Plane Method

Input: An integer program (IP) and its initial linear relaxation (P_0)

Output: If the procedure terminates, an optimal integral solution or a valid failure declaration

$i \leftarrow 0$;
while *no answer has been returned* **do**
    Solve the current linear relaxation (P_i);
    if (P_i) *is infeasible* **then**
        return (IP) *is infeasible;*
        ;
    end
    if (P_i) *is unbounded* **then**
        return (IP) *is infeasible or unbounded;*
        ;
    end
    if *the optimal solution* $\bar{\vec{x}}$ *of* (P_i) *is integral* **then**
        return $\bar{\vec{x}}$ *as an optimal solution for* (IP)*;*
        ;
    end
    Find a cutting plane $\vec{\alpha}^\top \vec{x} \le \beta$ that is valid for (IP) but violated by $\bar{\vec{x}}$;
    Set $(P_{i+1}) \leftarrow (P_i)$ augmented with $\vec{\alpha}^\top \vec{x} \le \beta$; set $i \leftarrow i+1$;
end

The remaining question is how to find such cuts systematically. This generic template does not by itself guarantee finite termination: that requires a particular cut family and a corresponding convergence theorem. The Gomory cuts developed next provide a systematic construction for pure integer programs.

Cutting Planes from Simplex Tableaux

Example 20.5 Return to Example 20.4. After adding slack variables, we obtain

$$\begin{aligned} \max \quad & \begin{bmatrix} 2 \\ 5 \\ 0 \\ 0 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 1 & 4 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 8 \\ 4 \end{bmatrix}, \\ & \vec{x} \geq \vec{0}, \end{aligned}$$

with the integrality restriction on x_1, x_2, x_3, x_4. Solving the linear relaxation by the simplex algorithm yields the optimal basis $\{1, 2\}$ and the canonical form

$$\begin{aligned} \max \quad & 12 + \begin{bmatrix} 0 \\ 0 \\ -1 \\ -1 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 1 & 0 & -1/3 & 4/3 \\ 0 & 1 & 1/3 & -1/3 \end{bmatrix} \vec{x} = \begin{bmatrix} 8/3 \\ 4/3 \end{bmatrix}, \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

The corresponding basic feasible solution is $\begin{bmatrix} 8/3 \\ 4/3 \\ 0 \\ 0 \end{bmatrix}$, so it is optimal for the relaxation.

The equation

$$x_1 - \frac{1}{3}x_3 + \frac{4}{3}x_4 = \frac{8}{3}$$

holds for every solution of the relaxation, and therefore for every integer solution as well. Taking floors gives the valid inequality

$$x_1 - x_3 + x_4 \leq 2.$$

Since the fractional optimum $\begin{bmatrix} 8/3 \\ 4/3 \\ 0 \\ 0 \end{bmatrix}$ violates this inequality, it is a cutting plane. Adding it produces the stronger relaxation

$$\begin{aligned} \max \quad & \begin{bmatrix} 2 \\ 5 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot \vec{x} \\ \text{subject to} \quad & \begin{bmatrix} 1 & 4 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 1 & 1 \end{bmatrix} \vec{x} = \begin{bmatrix} 8 \\ 4 \\ 2 \end{bmatrix}, \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

Lecture 21 — Tuesday, March 25, 2014

Example 21.1 (Continuation of the cutting plane example) A similar floor argument can be applied to the second row of the tableau as well.

Last time we obtained the cutting plane coming from the row

$$x_1 - \frac{1}{3}x_3 + \frac{4}{3}x_4 = \frac{8}{3}.$$

Applying the same idea to the second row,

$$x_2 + \frac{1}{3}x_3 - \frac{1}{3}x_4 = \frac{4}{3},$$

produces the additional cutting plane

$$x_2 - x_4 \leq 1.$$

For a pure integer program, in which the slack variables are also integral, the general cutting plane method is as follows. First solve the LP relaxation; that is, drop the integrality constraints and solve the resulting linear program by the simplex method. Once a basis B is obtained, rewrite the relaxation in canonical form with respect to B:

$$\begin{aligned} \max \quad & \bar{z} + \bar{c}_N^\top x_N \\ \text{subject to} \quad & x_B + \bar{A}_N x_N = \bar{b}, \\ & x \geq 0. \end{aligned}$$

The associated basic solution is obtained by setting the nonbasic variables equal to zero, so $\bar{x}_N = 0$ and $\bar{x}_B = \bar{b}$. If $\bar{b} \in \mathbb{Z}^m$, then this basic solution is integral, so the LP relaxation has already produced a solution to the integer program.

If some entry of $\bar{b}$ is fractional, then choose an index i with $\bar{b}_i \notin \mathbb{Z}$. The ith row of the canonical system has the form

$$x_{B_i} + \sum_{j \in N} \bar{A}_{ij} x_j = \bar{b}_i.$$

For every integer feasible solution, this equation implies the valid inequality

$$x_{B_i} + \sum_{j \in N} \lfloor \bar{A}_{ij} \rfloor x_j \leq \lfloor \bar{b}_i \rfloor.$$

This is the cutting plane determined by row i. At the current basic solution we have $x_N = 0$, so the left-hand side equals $x_{B_i} = \bar{b}_i$. Because $\bar{b}_i$ is fractional, we have $\bar{b}_i > \lfloor \bar{b}_i \rfloor$, so the present basic solution violates the inequality. Therefore the cut removes the current fractional LP solution while remaining valid for all integer feasible solutions.

6. Nonlinear Optimization

The final arc of the course focuses on convex nonlinear models, where strong structure restores tractability.

Nonlinear Optimization Can Be Hard

Recall that a **nonlinear program** (**NLP**) has the form

$$\begin{aligned} \min \quad & f(\vec{x}) \\ \text{subject to} \quad & g_i(\vec{x}) \leq 0 \qquad i = 1, \ldots, m. \end{aligned}$$

Linear programs are a special case of nonlinear programs, but general nonlinear optimization is *very hard.*

Definition 21.2 A **0–1 integer program** is an integer program in which every variable is constrained to lie in $\{0, 1\}$.

0–1 integer programs are intrinsically difficult. Moreover, solving nonlinear programs is at least as hard as solving 0–1 integer programs, because the constraint $x_j \in \{0, 1\}$ is equivalent to the nonlinear equation

$$x_j(1 - x_j) = 0.$$

Even small nonlinear programs can encode very hard questions.

Example 21.3 Consider

$$\begin{array}{rl} \min & (x_1^{x_4} + x_2^{x_4} - x_3^{x_4})^2 + (\sin \pi x_1)^2 + (\sin \pi x_2)^2 + (\sin \pi x_3)^2 + (\sin \pi x_4)^2 \\ \text{subject to} & 1 - x_i \le 0, \quad i = 1, 2, 3 \\ & 3 - x_4 \le 0 \end{array}.$$

Clearly the objective function is nonnegative for all feasible $\vec{x}$. A solution that achieves a value of 0 would force all of the squared terms to vanish. The sine terms would then imply $x_1, \ldots, x_4 \in \mathbb{Z}$, while the first term would encode an equation of the form

$$x_1^{x_4} + x_2^{x_4} = x_3^{x_4},$$

so a feasible point with objective value 0 would be a counterexample to Fermat's last theorem. Thus even deciding whether this NLP attains 0 encodes an extremely difficult Diophantine question.

Convexity

Because general nonlinear programs are so hard, the rest of the course focuses on convex settings. Consider the problem

$$\begin{array}{rl} \min & x_2 \\ \text{subject to} & \vec{x} \in S \subseteq \mathbb{R}^2 \end{array}$$

where S is the set shown in Figure 27.

The key feature of convex sets is that improvement directions can be turned into nearby better feasible points.

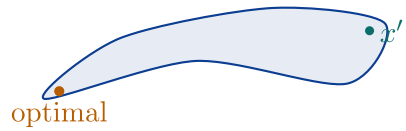


FIGURE 27

Proposition 21.4 Let $S \subseteq \mathbb{R}^n$ be convex, and let $\vec{x}^1, \vec{x}^2 \in S$. If $\vec{c}^\top \vec{x}^1 > \vec{c}^\top \vec{x}^2$, then there exists a point in S arbitrarily close to $\vec{x}^1$ whose objective value is strictly smaller than $\vec{c}^\top \vec{x}^1$.

Proof. For every $\lambda \in (0,1)$, the point

$$\vec{x}^\lambda = (1-\lambda)\vec{x}^1 + \lambda \vec{x}^2$$

belongs to S by convexity. Moreover,

$$\vec{c}^\top \vec{x}^\lambda = (1-\lambda)\vec{c}^\top \vec{x}^1 + \lambda \vec{c}^\top \vec{x}^2 < \vec{c}^\top \vec{x}^1.$$

As $\lambda \to 0$, the points $\vec{x}^\lambda$ approach $\vec{x}^1$. $\square$

Lecture 22 — Thursday, March 27, 2014

From this point onward we focus on nonlinear programs with linear objective functions and convex feasible regions.

Definition 22.1 A function $f : \mathbb{R}^n \to \mathbb{R}$ is **convex** if for every $\vec{a}, \vec{b} \in \mathbb{R}^n$ and every $\lambda \in [0,1]$,

$$f(\lambda \vec{a} + (1-\lambda)\vec{b}) \le \lambda f(\vec{a}) + (1-\lambda) f(\vec{b}).$$

Example 22.2 The function $f : \mathbb{R} \to \mathbb{R}$ given by $f(x) = x^2$ is convex.

Proof. Let $a, b \in \mathbb{R}$ and let $\lambda \in [0,1]$. Then

$$(\lambda a + (1-\lambda)b)^2 \le \lambda a^2 + (1-\lambda)b^2$$

is equivalent to

$$a^2 + b^2 - 2ab = (a - b)^2 \geq 0.$$

Hence x^2 is convex. □

Convex functions are useful because they produce convex feasible regions.

Proposition 22.3 If all of the functions $g_1, \ldots, g_m$ are convex, then the feasible region of

$$\begin{array}{rll} \min & \vec{c}^\top \vec{x} & \\ \text{subject to} & g_i(\vec{x}) \leq 0 & i = 1, \ldots, m \end{array}$$

is convex.

Proof. For each i, let

$$S_i := \{\vec{x} \mid g_i(\vec{x}) \leq 0\}.$$

If $\vec{a}, \vec{b} \in S_i$ and $\lambda \in [0, 1]$, then convexity of g_i gives

$$g_i(\lambda\vec{a} + (1 - \lambda)\vec{b}) \leq \lambda g_i(\vec{a}) + (1 - \lambda)g_i(\vec{b}) \leq 0.$$

Thus each S_i is convex. The feasible region is $\bigcap_{i=1}^m S_i$, so it is convex as well. □

This assumption is genuinely necessary. Without convexity of the constraint functions, the feasible region need not be convex.

Epigraphs, Subgradients, and Supporting Half-Spaces

These two notions are central in convex analysis because they turn nonlinear behavior into geometric and linear objects.

Definition 22.4 Let $f : \mathbb{R}^n \to \mathbb{R}$. The **epigraph** of f is

$$\text{epi}(f) := \left\{ \begin{bmatrix} y \\ \vec{x} \end{bmatrix} \in \mathbb{R} \times \mathbb{R}^n \mid f(\vec{x}) \leq y \right\}.$$

Geometrically, this is the region on or above the graph of f. Figure 28 depicts the graph of f together with its epigraph.

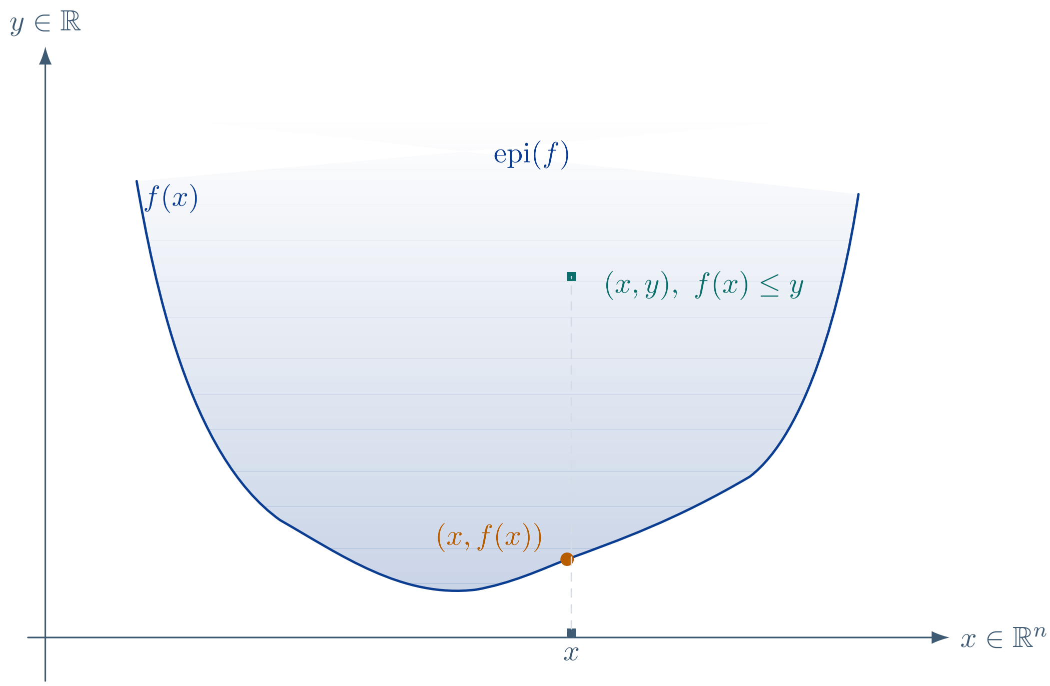


Figure 28

Theorem 22.5 (Epigraph characterization of convexity) A function f is convex if and only if epi(f) is convex.

Proof. *Forward implication.* Assume f is convex. Take any

$$\begin{bmatrix} y_1 \\ \vec{x}_1 \end{bmatrix}, \begin{bmatrix} y_2 \\ \vec{x}_2 \end{bmatrix} \in \text{epi}(f), \quad \lambda \in [0, 1].$$

By definition of epigraph, $f(\vec{x}_1) \leq y_1$ and $f(\vec{x}_2) \leq y_2$. Using convexity of f,

$$f(\lambda\vec{x}_1 + (1-\lambda)\vec{x}_2) \leq \lambda f(\vec{x}_1) + (1-\lambda)f(\vec{x}_2) \leq \lambda y_1 + (1-\lambda)y_2.$$

Hence

$$\begin{bmatrix} \lambda y_1 + (1-\lambda)y_2 \\ \lambda\vec{x}_1 + (1-\lambda)\vec{x}_2 \end{bmatrix} \in \text{epi}(f),$$

so epi(f) is convex.

Reverse implication. Assume $\operatorname{epi}(f)$ is convex. Let $\vec{x}_1, \vec{x}_2 \in \mathbb{R}^n$ and $\lambda \in [0,1]$. Then

$$\begin{bmatrix} f(\vec{x}_1) \\ \vec{x}_1 \end{bmatrix}, \begin{bmatrix} f(\vec{x}_2) \\ \vec{x}_2 \end{bmatrix} \in \operatorname{epi}(f).$$

By convexity of $\operatorname{epi}(f)$,

$$\lambda \begin{bmatrix} f(\vec{x}_1) \\ \vec{x}_1 \end{bmatrix} + (1-\lambda) \begin{bmatrix} f(\vec{x}_2) \\ \vec{x}_2 \end{bmatrix} = \begin{bmatrix} \lambda f(\vec{x}_1) + (1-\lambda) f(\vec{x}_2) \\ \lambda \vec{x}_1 + (1-\lambda)\vec{x}_2 \end{bmatrix} \in \operatorname{epi}(f).$$

Therefore,

$$f(\lambda \vec{x}_1 + (1-\lambda)\vec{x}_2) \le \lambda f(\vec{x}_1) + (1-\lambda) f(\vec{x}_2),$$

which is exactly convexity of f. □

The plan for the rest of the course is to replace selected nonlinear constraints by supporting linear inequalities. Starting from

$$\begin{aligned} \min \quad & \vec{c}^\top \vec{x} \\ \text{subject to} \quad & g_i(\vec{x}) \le 0 \qquad i = 1, \dots, m \end{aligned}$$

we will linearize some of the active constraints and thereby obtain relaxations with larger feasible regions.

Definition 22.6 Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and let $\bar{\vec{x}} \in \mathbb{R}^n$. A vector $\vec{s} \in \mathbb{R}^n$ is a **subgradient** of f at $\bar{\vec{x}}$ if

$$f(\bar{\vec{x}}) + \vec{s}^\top (\vec{x} - \bar{\vec{x}}) \le f(\vec{x}) \quad \text{for every } \vec{x} \in \mathbb{R}^n.$$

The affine function

$$\begin{aligned} h(\vec{x}) &:= f(\bar{\vec{x}}) + \vec{s}^\top (\vec{x} - \bar{\vec{x}}) \\ &= \vec{s}^\top \vec{x} - [\vec{s}^\top \bar{\vec{x}} - f(\bar{\vec{x}})] \end{aligned}$$

is therefore a global affine lower bound on f that is tight at $\bar{\vec{x}}$.

Example 22.7 Consider $g : \mathbb{R}^2 \to \mathbb{R}$ given by $g(\vec{x}) = x_2^2 - x_1$. We already know that g is convex. We claim that $\vec{s} = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$ is a subgradient of g at $\bar{\vec{x}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

Indeed,

$$h(\vec{x}) := g(\bar{\vec{x}}) + \vec{s}^\top(\vec{x} - \bar{\vec{x}}) = 0 + \begin{bmatrix} -1 & 2 \end{bmatrix} \left(\vec{x} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) = -x_1 + 2x_2 - 1.$$

We need to verify that $h(\vec{x}) \le g(\vec{x})$ for every $\vec{x} \in \mathbb{R}^2$. This is equivalent to

$$-x_1 + 2x_2 - 1 \le x_2^2 - x_1$$

or, after rearranging,

$$x_2^2 - 2x_2 + 1 = (x_2 - 1)^2 \ge 0,$$

which is clear.

Supporting half-spaces provide the geometric tool that turns subgradients into linear relaxations.

Recall the definitions of hyperplanes and half-spaces.

Definition 22.8 Let $C \subseteq \mathbb{R}^n$ be convex. We say that the half-space

$$F := \{\vec{x} \in \mathbb{R}^n \mid \vec{s}^\top \vec{x} \le \beta\}$$

is a **supporting half-space** of C at $\bar{\vec{x}}$ if:

(1) $C \subseteq F$;

(2) $\vec{s}^\top \bar{\vec{x}} = \beta$, that is, $\bar{\vec{x}}$ lies on the hyperplane defining the boundary of F.

Proposition 22.9 Let $g : \mathbb{R}^n \to \mathbb{R}$ be convex, let $\bar{\vec{x}} \in \mathbb{R}^n$ satisfy $g(\bar{\vec{x}}) = 0$, and let $\vec{s} \in \mathbb{R}^n \setminus \{\vec{0}\}$ be a nonzero subgradient of g at $\bar{\vec{x}}$. Denote by

$$C = \{\vec{x} \in \mathbb{R}^n \mid g(\vec{x}) \le 0\}$$

the 0-level set of g, and by

$$F = \{\vec{x} \in \mathbb{R}^n \mid g(\bar{\vec{x}}) + \vec{s}^\top(\vec{x} - \bar{\vec{x}}) \le 0\}$$

the half-space defined by the subgradient inequality. Then F is a supporting half-space of C at $\bar{\vec{x}}$.

Proof. Let $\vec{x} \in C$. Then $g(\vec{x}) \leq 0$. Since $\vec{s}$ is a subgradient of g at $\bar{\vec{x}}$,

$$g(\bar{\vec{x}}) + \vec{s}^\top(\vec{x} - \bar{\vec{x}}) \leq g(\vec{x}) \leq 0.$$

Thus $\vec{x} \in F$, so $C \subseteq F$. Also,

$$g(\bar{\vec{x}}) + \vec{s}^\top(\bar{\vec{x}} - \bar{\vec{x}}) = g(\bar{\vec{x}}) = 0,$$

so $\bar{\vec{x}}$ lies on the boundary of F. □

Example 22.10 For the function $g(\vec{x}) = x_2^2 - x_1$ and the point $\bar{\vec{x}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, the subgradient $\vec{s} = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$ gives the supporting half-space

$$F = \{\vec{x} \in \mathbb{R}^2 \mid -x_1 + 2x_2 - 1 \leq 0\}.$$

From Proposition 22.9 we obtain the following corollary:

Corollary 22.11 Consider an NLP of the form

$$\begin{aligned} \min \quad & \vec{c}^\top \vec{x} \\ \text{subject to} \quad & g_i(\vec{x}) \leq 0 \qquad i = 1, \ldots, m. \end{aligned}$$

Let $\bar{\vec{x}}$ be a feasible solution and suppose that the constraint $g_i(\vec{x}) \leq 0$ is tight at $\bar{\vec{x}}$. Suppose also that g_i is convex and has a subgradient $\vec{s}$ at $\bar{\vec{x}}$. Then the NLP obtained by replacing the constraint $g_i(\vec{x}) \leq 0$ with the linear constraint

$$\vec{s}^\top \vec{x} \leq \vec{s}^\top \bar{\vec{x}} - g_i(\bar{\vec{x}})$$

is a relaxation of the original NLP.

Proof. Let

$$C_i = \{\vec{x} \in \mathbb{R}^n \mid g_i(\vec{x}) \leq 0\}.$$

Since $\bar{\vec{x}}$ is tight for the ith constraint, we have $g_i(\bar{\vec{x}}) = 0$. If $\vec{x} \in C_i$, the subgradient inequality gives

$$g_i(\bar{\vec{x}}) + \vec{s}^\top(\vec{x} - \bar{\vec{x}}) \leq g_i(\vec{x}) \leq 0.$$

Rearranging gives

$$\vec{s}^\top \vec{x} \leq \vec{s}^\top \bar{\vec{x}} - g_i(\bar{\vec{x}}).$$

Thus every point satisfying the original constraint $g_i(\vec{x}) \leq 0$ also satisfies the new linear constraint, even in the degenerate case $\vec{s} = \vec{0}$. Therefore, if we replace the ith constraint of the NLP by this linear inequality and leave all other constraints unchanged, every feasible solution of the original NLP remains feasible for the new problem. Hence the new NLP is a relaxation of the original NLP. $\square$

Lecture 23 — Tuesday, April 01, 2014

Optimality Conditions for Convex Programs

We now derive practical optimality criteria for convex programs using gradients, cones, and multiplier conditions. Consider convex NLPs of the form

$$\begin{aligned} \min \quad & f(\vec{x}) \\ \text{subject to} \quad & g_1(\vec{x}) \leq 0, \\ & \vdots \\ & g_m(\vec{x}) \leq 0. \end{aligned}$$

It is enough to understand the case where the objective function is linear. Indeed, by introducing a new variable x_{n+1} and the extra constraint $f(\vec{x}) - x_{n+1} \leq 0$, we can rewrite the problem as

$$\begin{aligned} \min \quad & x_{n+1} \\ \text{subject to} \quad & f(\vec{x}) - x_{n+1} \leq 0, \\ & g_1(\vec{x}) \leq 0 \\ & \vdots \\ & g_m(\vec{x}) \leq 0. \end{aligned}$$

If $\vec{x}'$ is a Slater point for the original problem, then choosing $x'_{n+1} > f(\vec{x}')$ makes every constraint in this epigraph formulation strict, so the Slater condition is preserved.

We therefore consider problems with linear objectives of the form

$$\begin{aligned} \min \quad & \vec{c}^\top \vec{x} \\ \text{subject to} \quad & g_1(\vec{x}) \le 0, \\ & \vdots \\ & g_m(\vec{x}) \le 0. \end{aligned}$$

Proposition 23.1 Let $\bar{\vec{x}}$ be feasible for

$$\begin{aligned} \min \quad & \vec{c}^\top \vec{x} \\ \text{subject to} \quad & g_1(\vec{x}) \le 0, \\ & \vdots \\ & g_m(\vec{x}) \le 0. \end{aligned}$$

Let

$$J(\bar{\vec{x}}) := \{i \mid g_i(\bar{\vec{x}}) = 0\}$$

be the set of tight constraints at $\bar{\vec{x}}$. Suppose that for every $i \in J(\bar{\vec{x}})$, the function g_i has a subgradient $\vec{s}_i$ at $\bar{\vec{x}}$. If

$$-\vec{c} \in \operatorname{cone}\{\vec{s}_i \mid i \in J(\bar{\vec{x}})\},$$

then $\bar{\vec{x}}$ is an optimal solution.

Proof. For each $i \in J(\bar{\vec{x}})$, the preceding discussion shows that the linear constraint

$$\vec{s}_i^\top \vec{x} \le \vec{s}_i^\top \bar{\vec{x}} - g_i(\bar{\vec{x}})$$

defines a relaxation of the nonlinear constraint $g_i(\vec{x}) \le 0$. Since $g_i(\bar{\vec{x}}) = 0$ for every $i \in J(\bar{\vec{x}})$, we obtain the LP relaxation

$$\begin{aligned} \min \quad & \vec{c}^\top \vec{x} \\ \text{subject to} \quad & \vec{s}_i^\top \vec{x} \le \vec{s}_i^\top \bar{\vec{x}} \qquad i \in J(\bar{\vec{x}}). \end{aligned}$$

This is indeed a relaxation of the original NLP, because each tight nonlinear constraint has been replaced by a valid linear relaxation, and the remaining constraints have simply been dropped.

By hypothesis,

$$-\vec{c} \in \operatorname{cone}\{\vec{s}_i \mid i \in J(\bar{\vec{x}})\},$$

so there exist scalars $\lambda_i \geq 0$ for $i \in J(\bar{\vec{x}})$ such that

$$-\vec{c} = \sum_{i \in J(\bar{\vec{x}})} \lambda_i \vec{s}_i.$$

Now let $\vec{x}$ be any feasible solution of the LP relaxation. Then for every $i \in J(\bar{\vec{x}})$ we have

$$\vec{s}_i^\top \vec{x} \leq \vec{s}_i^\top \bar{\vec{x}}.$$

Therefore,

$$\vec{c}^\top \vec{x} - \vec{c}^\top \bar{\vec{x}} = \vec{c}^\top (\vec{x} - \bar{\vec{x}}) = -\sum_{i \in J(\bar{\vec{x}})} \lambda_i \vec{s}_i^\top (\vec{x} - \bar{\vec{x}}) = \sum_{i \in J(\bar{\vec{x}})} \lambda_i \big(\vec{s}_i^\top \bar{\vec{x}} - \vec{s}_i^\top \vec{x}\big) \geq 0.$$

Hence $\vec{c}^\top \bar{\vec{x}} \leq \vec{c}^\top \vec{x}$ for every feasible solution $\vec{x}$ of the LP relaxation, so $\bar{\vec{x}}$ is optimal for that LP. Since $\bar{\vec{x}}$ is feasible for the original NLP and the LP is a relaxation, every feasible solution of the NLP is feasible for the LP. It follows that no feasible point of the NLP can have smaller objective value than $\bar{\vec{x}}$. Therefore $\bar{\vec{x}}$ is optimal for the original NLP. $\square$

This gives sufficient conditions for optimality: if the negative objective vector $-\vec{c}$ lies in the cone generated by the subgradients of the tight constraints, then the point is optimal. We now turn to necessary conditions.

Differentiability and Gradients

We start off by reviewing standard definitions from multivariable calculus.

Definition 23.2

Let $f : \mathbb{R}^n \to \mathbb{R}$ and let $\bar{\vec{x}} \in \mathbb{R}^n$. If there exists $\vec{s} \in \mathbb{R}^n$ such that

$$\lim_{\vec{h} \to \vec{0}} \frac{f(\bar{\vec{x}} + \vec{h}) - f(\bar{\vec{x}}) - \vec{s}^\top \vec{h}}{\|\vec{h}\|_2} = 0,$$

then we say that f is **differentiable** at $\bar{\vec{x}}$ and call the vector $\vec{s}$ the **gradient** of f at $\bar{\vec{x}}$. We denote it by $\nabla f(\bar{\vec{x}})$.

Under the usual assumptions, the gradient is given by

$$\nabla f(\bar{\vec{x}}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\bar{\vec{x}}) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\bar{\vec{x}}) \end{bmatrix}.$$

Proposition 23.3 Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and let $\bar{\vec{x}} \in \mathbb{R}^n$. If the gradient $\nabla f(\bar{\vec{x}})$ exists, then it is a subgradient of f at $\bar{\vec{x}}$.

Proof. Let $\vec{x} \in \mathbb{R}^n$ be arbitrary, and define

$$\phi(t) := f\big(\bar{\vec{x}} + t(\vec{x} - \bar{\vec{x}})\big) \quad \text{for } t \in [0,1].$$

Since f is convex, the function ϕ is convex on $[0,1]$. The slopes of its secant lines are nondecreasing, so its right derivative at 0 satisfies

$$\phi'(0) \leq \frac{\phi(1) - \phi(0)}{1 - 0}.$$

Now $\phi(1) = f(\vec{x})$ and $\phi(0) = f(\bar{\vec{x}})$. Because f is differentiable at $\bar{\vec{x}}$, the chain rule gives

$$\phi'(0) = \nabla f(\bar{\vec{x}})^\top (\vec{x} - \bar{\vec{x}}).$$

Substituting into the previous inequality yields

$$f(\vec{x}) \geq f(\bar{\vec{x}}) + \nabla f(\bar{\vec{x}})^\top (\vec{x} - \bar{\vec{x}}).$$

Since this holds for every $\vec{x} \in \mathbb{R}^n$, the vector $\nabla f(\bar{\vec{x}})$ satisfies the defining inequality for a subgradient of f at $\bar{\vec{x}}$. Thus $\nabla f(\bar{\vec{x}})$ is a subgradient of f at $\bar{\vec{x}}$. □

Example 23.4 Consider again the function $g(\vec{x}) = x_2^2 - x_1$ from Example 22.7. Its gradient is

$$\nabla g(\vec{x}) = \begin{bmatrix} -1 \\ 2x_2 \end{bmatrix}.$$

Evaluating at $\bar{\vec{x}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ gives

$$\nabla g(\bar{\vec{x}}) = \begin{bmatrix} -1 \\ 2 \end{bmatrix}.$$

Since g is convex, Proposition 23.3 tells us that this gradient is also a subgradient at $\bar{\vec{x}}$.

Karush–Kuhn–Tucker Conditions

We now assemble Slater feasibility, convexity, and gradients into an optimality criterion for NLPs.

Definition 23.5 We need one more notion. We say that the NLP has a **Slater point** if there exists $\vec{x}'$ such that

$$g_i(\vec{x}') < 0 \quad \text{for all } i = 1, \ldots, m.$$

That is, there is a point satisfying every inequality strictly.

Theorem 23.6 (Karush–Kuhn–Tucker theorem based on gradients) Consider a convex NLP of the form

$$\begin{aligned} \min \quad & f(\vec{x}) \\ \text{subject to} \quad & g_1(\vec{x}) \le 0, \\ & \vdots \\ & g_m(\vec{x}) \le 0 \end{aligned}$$

that has a Slater point. Let $\bar{\vec{x}} \in \mathbb{R}^n$ be a feasible solution and assume that $f, g_1, \ldots, g_m$ are differentiable at $\bar{\vec{x}}$. Then $\bar{\vec{x}}$ is an optimal solution if and only if

$$-\nabla f(\bar{\vec{x}}) \in \operatorname{cone}\{\nabla g_i(\bar{\vec{x}}) \mid i \in J(\bar{\vec{x}})\},$$

where $J(\bar{\vec{x}}) = \{i \mid g_i(\bar{\vec{x}}) = 0\}$.

When $f(\vec{x}) = \vec{c}^\top \vec{x}$, the sufficiency direction is exactly the previous proposition. The converse direction is harder and needs the Slater condition; it is established below using Farkas' lemma.

The condition in Theorem 23.6 can also be restated in terms of multipliers: there exists $\vec{y} \in \mathbb{R}^m_+$ such that

$$\nabla f(\bar{\vec{x}}) + \sum_{i=1}^{m} y_i \nabla g_i(\bar{\vec{x}}) = 0$$

and

$$y_i g_i(\bar{\vec{x}}) = 0 \quad \text{for } i = 1, \ldots, m.$$

This is the nonlinear analogue of dual feasibility together with complementary slackness.

To prove the KKT theorem in more generality, we will require Farkas' lemma, of which there are many equivalent variants.

Theorem 23.7 (Farkas' lemma) Let $\vec{a}^{(1)}, \ldots, \vec{a}^{(m)} \in \mathbb{R}^n$, and let $\vec{b} \in \mathbb{R}^n$. Then exactly one of the following is true:

(i) $\vec{b} \in \operatorname{cone}\{\vec{a}^{(i)} \mid i = 1, \ldots, m\}$.

(ii) There exists $\vec{y} \in \mathbb{R}^n$ such that

$$\vec{y}^\top \vec{a}^{(i)} \geq 0 \quad \text{for } i = 1, \ldots, m, \qquad \text{and} \qquad \vec{y}^\top \vec{b} < 0.$$

Equivalently, the hyperplane $H := \{\vec{h} \in \mathbb{R}^n \mid \vec{y}^\top \vec{h} = 0\}$ separates $\vec{b}$ from the cone generated by $\vec{a}^{(1)}, \ldots, \vec{a}^{(m)}$.

We will prove an equivalent version later.

Lecture 24 — Thursday, April 03, 2014

We continue with the same Karush–Kuhn–Tucker criterion, now stated in the linear objective form used most often in applications.

Theorem 24.1 (Karush–Kuhn–Tucker theorem) Consider the nonlinear program

$$\begin{aligned} \min \quad & \vec{c}^\top \vec{x} \\ \text{subject to} \quad & g_i(\vec{x}) \leq 0 \qquad i = 1, \ldots, m \end{aligned}$$

and let $\bar{\vec{x}}$ be a feasible point. Suppose that all of the functions g_i are convex, all of the functions g_i are differentiable at $\bar{\vec{x}}$, and the problem has a Slater point. Then the following are equivalent:

1. $\bar{\vec{x}}$ is optimal.
2. $$-\vec{c} \in \operatorname{cone}\{\nabla g_i(\bar{\vec{x}}) \mid i \in J(\bar{\vec{x}})\},$$
   where $J(\bar{\vec{x}}) = \{i \mid g_i(\bar{\vec{x}}) = 0\}$.

Proof. We first show that (2) implies (1). Since each g_i is convex and differentiable at $\bar{\vec{x}}$, Proposition 23.3 implies that $\nabla g_i(\bar{\vec{x}})$ is a subgradient of g_i at $\bar{\vec{x}}$. Therefore the previous proposition applies with

$$\vec{s}_i = \nabla g_i(\bar{\vec{x}}) \quad \text{for } i \in J(\bar{\vec{x}}).$$

The hypothesis in (2) is exactly the cone condition from that proposition, so it follows that $\bar{\vec{x}}$ is optimal.

For the converse, assume that (1) holds, and suppose for contradiction that (2) fails. Then

$$-\vec{c} \notin \operatorname{cone}\{\nabla g_i(\bar{\vec{x}}) \mid i \in J(\bar{\vec{x}})\}.$$

By Farkas' lemma, there exists a vector $\vec{d} \in \mathbb{R}^n$ such that

$$\nabla g_i(\bar{\vec{x}})^\top \vec{d} \leq 0 \quad \text{for all } i \in J(\bar{\vec{x}}),$$

and

$$\vec{c}^\top \vec{d} < 0.$$

Let $\vec{x}'$ be a Slater point, so $g_i(\vec{x}') < 0$ for every $i = 1, \ldots, m$. For each $i \in J(\bar{\vec{x}})$, the vector $\nabla g_i(\bar{\vec{x}})$ is a subgradient of g_i at $\bar{\vec{x}}$, so

$$g_i(\vec{x}') \geq g_i(\bar{\vec{x}}) + \nabla g_i(\bar{\vec{x}})^\top (\vec{x}' - \bar{\vec{x}}).$$

Since $g_i(\bar{\vec{x}}) = 0$ for $i \in J(\bar{\vec{x}})$ and $g_i(\vec{x}') < 0$, this shows that

$$\nabla g_i(\bar{\vec{x}})^\top (\vec{x}' - \bar{\vec{x}}) < 0 \quad \text{for all } i \in J(\bar{\vec{x}}).$$

Because $\vec{c}^\top \vec{d} < 0$, we can choose $K > 0$ so large that

$$\vec{c}^\top \big((\vec{x}' - \bar{\vec{x}}) + K\vec{d}\big) < 0.$$

Set

$$\vec{v} := (\vec{x}' - \bar{\vec{x}}) + K\vec{d}.$$

Then for every $i \in J(\bar{\vec{x}})$,

$$\nabla g_i(\bar{\vec{x}})^\top \vec{v} = \nabla g_i(\bar{\vec{x}})^\top (\vec{x}' - \bar{\vec{x}}) + K \nabla g_i(\bar{\vec{x}})^\top \vec{d} < 0,$$

because the first term is strictly negative and the second is nonpositive.

Now consider the points

$$\vec{x}(t) := \bar{\vec{x}} + t\vec{v}, \quad t > 0.$$

For each active constraint $i \in J(\bar{\vec{x}})$, differentiability of g_i at $\bar{\vec{x}}$ gives

$$g_i(\vec{x}(t)) = g_i(\bar{\vec{x}}) + t\nabla g_i(\bar{\vec{x}})^\top \vec{v} + o(t).$$

Since $g_i(\bar{\vec{x}}) = 0$ and $\nabla g_i(\bar{\vec{x}})^\top \vec{v} < 0$, we obtain $g_i(\vec{x}(t)) < 0$ for all sufficiently small $t > 0$. For each inactive constraint $i \notin J(\bar{\vec{x}})$, we have $g_i(\bar{\vec{x}}) < 0$, so by continuity $g_i(\vec{x}(t)) < 0$ for all sufficiently small $t > 0$ as well. Hence $\vec{x}(t)$ is feasible for the NLP whenever $t > 0$ is small enough.

On the other hand,

$$\vec{c}^\top \vec{x}(t) = \vec{c}^\top \bar{\vec{x}} + t\vec{c}^\top \vec{v} < \vec{c}^\top \bar{\vec{x}}$$

for every sufficiently small $t > 0$, because $\vec{c}^\top \vec{v} < 0$. This contradicts the optimality of $\bar{\vec{x}}$. Therefore (2) must hold. □

The linear-objective theorem also completes the proof of Theorem 23.6. Apply it to the epigraph formulation at $(\bar{\vec{x}}, f(\bar{\vec{x}}))$. The gradient of the epigraph constraint $f(\vec{x}) - x_{n+1} \leq 0$ is $(\nabla f(\bar{\vec{x}}), -1)$, while each original active constraint has gradient $(\nabla g_i(\bar{\vec{x}}), 0)$. The last coordinate of the KKT cone equation forces the multiplier of the epigraph constraint to equal 1; the first n coordinates then give

$$-\nabla f(\bar{\vec{x}}) \in \operatorname{cone}\{\nabla g_i(\bar{\vec{x}}) : i \in J(\bar{\vec{x}})\}.$$

Conversely, any such cone representation, together with multiplier 1 for the epigraph constraint, satisfies the linear-objective KKT condition. Thus the two formulations have exactly the same optimality criterion.

Example 24.2 As a concrete application of the criterion, consider the problem

$$\begin{aligned} \min \quad & -7x_1 - 5x_2 \\ \text{subject to} \quad & 2x_1^2 + x_2^2 + x_1x_2 - 4 \leq 0, \\ & x_1^2 + x_2^2 - 2 \leq 0, \\ & -x_1 + \frac{1}{2} \leq 0. \end{aligned}$$

We show that $\bar{\vec{x}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ is optimal.

The constraint functions are differentiable and convex, and a Slater point is given by $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$. At $\bar{\vec{x}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, the first two constraints are tight, while the third is not. Their gradients are

$$\nabla g_1(1,1) = \begin{bmatrix} 5 \\ 3 \end{bmatrix} \qquad \text{and} \qquad \nabla g_2(1,1) = \begin{bmatrix} 2 \\ 2 \end{bmatrix}.$$

Therefore

$$-\vec{c} = \begin{bmatrix} 7 \\ 5 \end{bmatrix} = 1 \cdot \begin{bmatrix} 5 \\ 3 \end{bmatrix} + 1 \cdot \begin{bmatrix} 2 \\ 2 \end{bmatrix},$$

so $-\vec{c}$ lies in the cone generated by the gradients of the tight constraints. The KKT criterion therefore proves that $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ is optimal.

Farkas' Lemma

Farkas' lemma also has a clean geometric interpretation.

In $\mathbb{R}^2$, the statement takes the geometric form shown in Figure 29.

This formulation is important because it turns an algebraic alternative theorem into a geometric separation statement.

For computation and duality arguments, it is useful to rewrite the geometric statement in matrix form. Another equivalent formulation is as follows:

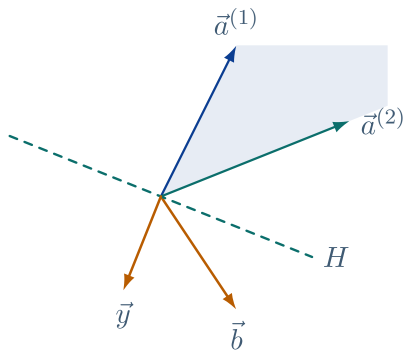


FIGURE 29

Theorem 24.3 (Farkas' lemma: second version) Let $A = [\vec{a}^{(1)} \ \dots \ \vec{a}^{(m)}]$. Then exactly one of the following holds:

(i) The system $A\vec{x} = \vec{b}$, $\vec{x} \geq \vec{0}$ has a solution.

(ii) There exists $\vec{y} \in \mathbb{R}^n$ such that $\vec{y}^\top A \geq 0$ and $\vec{y}^\top \vec{b} < 0$.

Proof. First note that ((i)) and ((ii)) cannot both hold. Indeed, if $A\vec{x} = \vec{b}$ with $\vec{x} \geq 0$ and if $\vec{y}^\top A \geq 0$, then

$$\vec{y}^\top \vec{b} = \vec{y}^\top A\vec{x} \geq 0,$$

which contradicts $\vec{y}^\top \vec{b} < 0$.

Now assume that ((i)) fails. Consider the linear program

$$\begin{aligned} \min \quad & 0 \\ \text{subject to} \quad & A\vec{x} = \vec{b}, \\ & \vec{x} \geq \vec{0}. \end{aligned}$$

Its dual is

$$\begin{aligned} \max \quad & \vec{y}^\top \vec{b} \\ \text{subject to} \quad & \vec{y}^\top A \leq 0, \\ & \vec{y} \text{ free}. \end{aligned}$$

The primal problem is infeasible by assumption. The dual is feasible, since $\vec{y} = \vec{0}$ satisfies $\vec{y}^\top A \leq 0$. If the dual had an optimal solution, applying strong duality after putting this dual problem in SEF—equivalently, applying strong duality with the two members of the dual pair interchanged—

would force the original primal to have an optimal solution. This contradicts infeasibility. The feasible dual therefore has no optimum, so the fundamental theorem of linear programming implies that it is unbounded. Hence there exists some $\vec{y}$ with

$$\vec{y}^\top A \leq 0 \qquad \text{and} \qquad \vec{y}^\top \vec{b} > 0.$$

Replacing $\vec{y}$ by $-\vec{y}$, we obtain a vector satisfying

$$(-\vec{y})^\top A \geq 0 \qquad \text{and} \qquad (-\vec{y})^\top \vec{b} < 0,$$

which is exactly ((ii)).

Thus, if ((i)) does not hold, then ((ii)) does. Since they cannot both hold, exactly one of them is true. $\square$

In fact, Farkas' lemma can be used to prove linear programming duality.

Appendix: Lecture and Tutorial Index