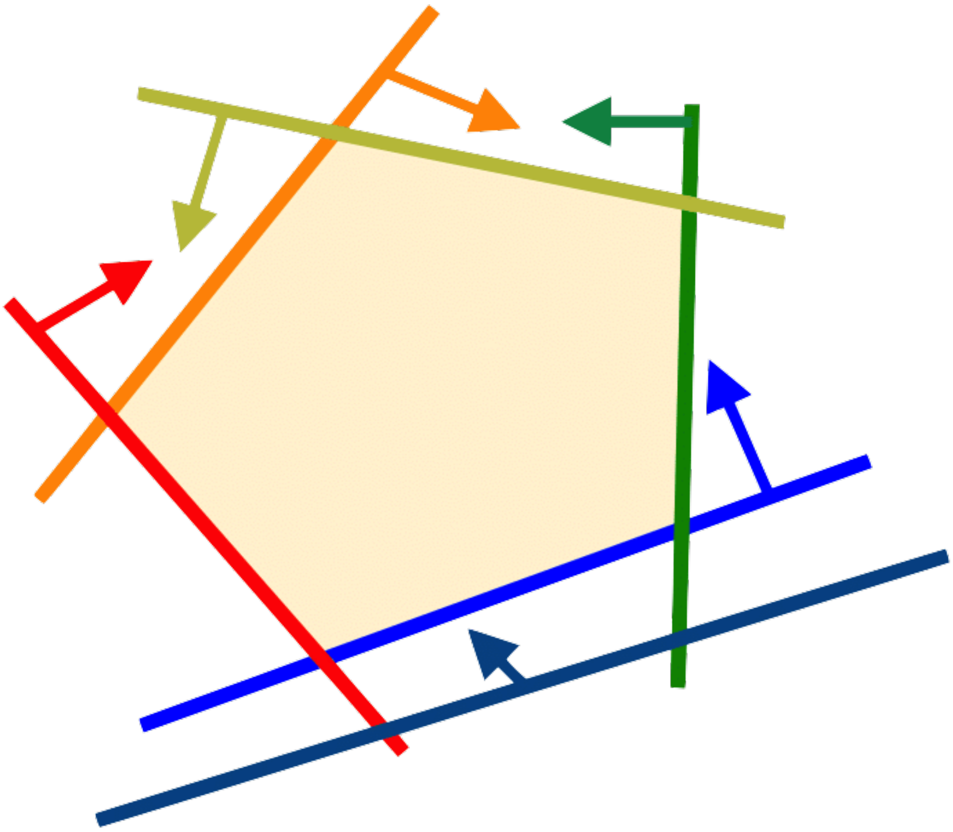
MATH561/IOE510/TO518
Linear Programming

Pingbang Hu

March 27, 2023

Abstract

This is the first course in the series of graduate-level, large-scale and rigorous mathematical programming courses taught by Jon Lee, and in particular, we will use the book write by professor Lee [Lee22]. This is a dynamic book which may changes and update constantly. In this course, we focus on developing a rigorous understanding on large-scale linear programming problems and also introduce the basic concept of integer programming.



This course is taken in Fall 2021, and the date on the covering page is the last updated time.

Contents

Chapter 1

Introduction to Linear Programming

Lecture 1: Introduction

1.1 General Linear Programming Problem

30 Aug. 08:00

Let's start with the definition of a linear programming problem.

Definition 1.1.1 (General linear programming problem)**.** A *general linear programming problem* is to either minimize or maximize an objective function with the constraints defined as follows.

Definition 1.1.2 (Objective function)**.** An *objective function* is in the form of

$$c_1x_1 + c_2x_2 + \cdots + c_nx_n,$$

where $c_i, x_i \in \mathbb{R}$, and x_i are variables for $i = 1, \ldots, n$.

Definition 1.1.3 (Constraint)**.** The *constraints* are the combination of structured constraints and also the signed constraints.

Definition 1.1.4 (Structured constraint)**.** The *structured constraints* are in the form of

$$a_{j1}x_1 + \cdots + a_{jn}x_n \gtreqqless b_j$$

where $a_{ji} \in \mathbb{R}$ for all $i = 1, \ldots, n$ and $j = 1, \ldots, m$.

Definition 1.1.5 (Signed constraint)**.** The *signed constraints* are in the form of

$$x_i \gtreqqless 0$$

for some $i = 1, \ldots, n$.

Notation. We often referred $\gtreqqless$ to either $\geq, \leq$ or $=$.

Remark. For a linear programming problem, we often assume there are n variables indexed by i, and m structured constraints indexed by j, and also signed constraints for some x_i.

Given a general linear programming problem, we have the following definitions.

Definition 1.1.6 (Solution)**.** We called an assignment of values to variable x as a *solution*.

Definition 1.1.7 (Feasible solution)**.** If this solution satisfies the linear constraints, we say that this solution is a *feasible solution*.

Definition 1.1.8 (Feasible region)**.** The set of feasible solutions is called the *feasible region*.

Definition 1.1.9 (Optimal solution)**.** A solution is *optimal* if there is no feasible solution with better objective value.

Remark. A feasible region is a polyhedron for a general linear programming problem.

1.1.1 Standard Form

It's convenient to group the coefficients together as

$$c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix},$$

since in this way, we can define the standard form of a linear programming.

Definition 1.1.10 (Standard form)**.** The *standard form* linear programming has the form of

$$\begin{aligned} \min\ & c^\top x \\ & Ax = b \\ & x \geq 0 \end{aligned}$$

with the condition that rows of A are linear independent.

Remark (Compactness of the solution space)**.** We only consider finitely many of constrains since with finite dimensional solution space will be compact, hence objective function can attain its extremum.

Remark (Convert to standard form)**.** Every general linear programming problem can be converted to standard form.

Proof. Given a general linear programming problem with our notation, we see that

- Sign:
  - If $x_i \leq 0 \Rightarrow x_i \to -x_i^-$, where $x_i^- \geq 0$.
  - If x_i is unrestricted $\Rightarrow x_i \to x_i^+ - x_i^-$, where $x_i^\pm \geq 0$.
- Constraints:
  - $\sum_{i=1}^n a_{ji}x_i \leq b_j \Rightarrow \sum_{i=1}^n a_{ji}x_i + s_j = b_j$, where $s_j \geq 0$.
  - $\sum_{i=1}^n a_{ji}x_i \geq b_j \Rightarrow \sum_{i=1}^n a_{ji}x_i - s_j = b_j$, where $s_j \geq 0$.
- Maximize: $\max \sum c_j x_j \Rightarrow -\min -\sum c_j x_j$.

⊛

We see that in order to convert to standard form, we sometimes need to introduce variable s_j for constraint j. This leads to the following.

Definition. Consider the new variable s_j introduced for j^{th} non-equal constraint when converting to the standard form.

Definition 1.1.11 (Slack variable). The *slack variable* s_j is introduced for $\leq$ constraint.

Definition 1.1.12 (Surplus variable). The *surplus variable* s_j is introduced for $\geq$ constraint.

Lecture 2: Duality

1.2 First Glance of Duality

1 Sep. 08:00

By looking at the standard form, another natural linear programming problem called the dual of the original problem arises.

Definition 1.2.1. Given a standard form linear programming problem called primal (P), the dual (D) of (P) is the following induced problem.

$$
\begin{array}{llll}
 & \min\ c^\top x & & \max\ y^\top b \\
 & \quad Ax = b & & \quad y^\top A \leq c^\top. \\
(\mathrm{P}) & \quad x \geq 0, & (\mathrm{D}) &
\end{array}
$$

Definition 1.2.2 (Primal). The problem (P) is the *primal*.

Definition 1.2.3 (Dual). The problem D is the *dual* of the primal (P).

Note. The dual is equivalent to

$$
\begin{aligned}
\max\ & b^\top y \\
& A^\top y \leq c.
\end{aligned}
$$

Then we have a direct, but important theorem.

Theorem 1.2.1 (Weak duality theorem). If $\hat{x}$ is feasible for (P) and $\hat{y}$ is feasible for (D), then

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

Proof. Since we have

$$
\hat{y}^\top A \leq c^\top \underset{\hat{x}\geq 0}{\Rightarrow} \hat{y}^\top A\hat{x} \leq \hat{c}^\top \hat{x} \underset{A\hat{x}=b}{\Rightarrow} \hat{y}^\top b \leq c^\top \hat{x},
$$

the result follows. ∎

Problem. Consider

$$
\begin{aligned}
\min\ & c^\top x \\
& Ax \geq b,
\end{aligned}
$$

turn this into the standard form problem and find the dual.

Answer. We see that x is unrestricted. We first minus a surplus variable s, we have

$$\begin{aligned} \min\ & c^\top x \\ & Ax - S = b \\ & s \geq 0. \end{aligned}$$

Now, we turn x into $x^+ - x^-$, namely

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \qquad x^+ := \begin{pmatrix} x_1^+ \\ \vdots \\ x_n^+ \end{pmatrix}, \qquad x^- := \begin{pmatrix} x_1^- \\ \vdots \\ x_n^- \end{pmatrix},$$

with $x^\pm \geq \vec{0}$. Then the original problem becomes

$$\begin{aligned} \min\ & c^\top (x^+ - x^-) \\ & A(x^+ - x^-) - s = b \\ & x^+, x^-, s \geq 0, \end{aligned}$$

equivalently,

$$\begin{aligned} \min\ & \begin{pmatrix} c^\top & -c^\top & 0 \end{pmatrix} \begin{pmatrix} x^+ \\ x^- \\ s \end{pmatrix} \\ & \begin{pmatrix} A & -A & -I \end{pmatrix} \begin{pmatrix} x^+ \\ x^- \\ s \end{pmatrix} = b \\ & \begin{pmatrix} x^+ \\ x^- \\ s \end{pmatrix} \geq 0. \end{aligned}$$

Set the dual variable being y, we further have

$$\begin{aligned} \max\ & y^\top b \\ & y^\top \begin{pmatrix} A & -A & -I \end{pmatrix} \leq \begin{pmatrix} c^\top & -c^\top & 0^\top \end{pmatrix}. \end{aligned}$$

⊛

Exercise. Show that the dual of the dual is the primal.

Lecture 3: Production Problem and Norm Minimization

1.3 Modeling

8 Sep. 08:00

We now see two main examples illustrating why linear programming is interesting to study.

1.3.1 Production Problem

The *production problem* can be formulated as follows.

$$\begin{aligned} \max\ & c^\top x \\ & Ax \leq b \\ & x \geq \vec{0}, \end{aligned}$$

where

- n products activities.

- c_j = per-unit revenue for activity $j = 1 \ldots n$.
- b_i = resource endowment for resource $i = 1 \ldots m$.
- a_{ij} = amount of resource i consumed by activity j.

> **Note.** This is a very important and practical problems considered all across the industry!

1.3.2 Norm Minimization

It's also useful to minimizing the norm of the variable x. And interestingly, this can be done by linear programming problems as well. Let's first consider minimizing a $\|\cdot\|_\infty$.

> **Definition 1.3.1** (Maximum norm). The *maximum norm* of $x \in \mathbb{R}^n$ is defined as
> $$\|x\|_\infty := \max_{1 \leq i \leq n} |x_i|.$$

Consider

$$\begin{aligned} \min \ & \|x\|_\infty \\ & Ax = b, \end{aligned}$$

we set up

$$\begin{aligned} \min \ & t \\ & t \geq x_i, \text{ for } i = 1, \ldots, n \\ & t \leq x_i, \text{ for } i = 1, \ldots, n \\ & Ax = b. \end{aligned}$$

We see that this optimization **pressure** will force the maximum of $|x_i|$ being small, hence we'll get the minimum among $|x_i|$. Similarly, we can consider the following minimizing $\|\cdot\|_1$.

> **Definition 1.3.2** (1-norm). The 1-*norm* of $x \in \mathbb{R}^n$ is defined as
> $$\|x\|_1 := \sum_{i=1}^n |x_i|.$$

> **Note** (p-norm). More generally, the p-*norm* of $x \in \mathbb{R}^n$ is defined as
> $$\|x\|_p := \left(\sum_{i=1}^n |x_i|^p \right)^{1/p},$$
> and $\|\cdot\|_1$, and even $\|\cdot\|_\infty$ are both special cases when $p = 1$ and $p = \infty$, respectively.

Consider

$$\begin{aligned} \min \ & \|x\|_1 \\ & Ax = b, \end{aligned}$$

we set up

$$\begin{aligned} \min \ & \sum_{i=1}^n t_i \\ & t_i \geq x_i, \quad \text{for } i = 1, \ldots, n \\ & t_i \leq -x_i, \text{ for } i = 1, \ldots, n \\ & Ax = b. \end{aligned}$$

Again, we see that the optimization pressure will force t_i goes to $|x_i|$, resulting $\sum_{i=1}^n t_i$ being $\|x\|_1$.

Remark. Minimize $\|x\|_1$ tends to make x **sparse** (lots of zeros).

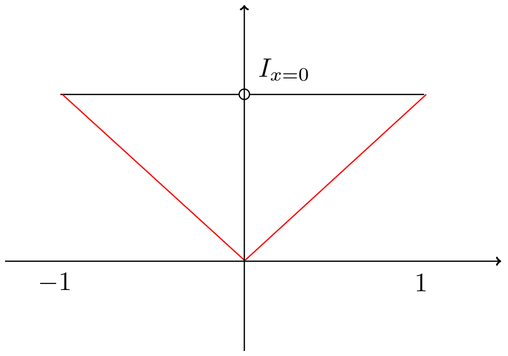


Figure 1.1: The best approximated convex function of $I_{x=0}$

Chapter 2

Algebra and Geometry

Lecture 4: Basis Partition

In this section, we're going to study various of fundamental building blocks for designing a mathematical complete algorithm for solving a standard form linear programming problem. This requires both algebraic and geometric understanding of the problem. Let's start with algebraic tools we need. 13 Sep. 08:00

2.1 Elementary Row Operations

There are some simple operations we can apply to a matrix called elementary row operations.

Definition 2.1.1 (Elementary row operations)**.** The following are called *elementary row operations.*

(a) Permute rows.

(b) Multiply a row by a non-zero factor.

(c) Add a multiple of a row to another row.

(d) Permutation in columns.

Note (Permutation in columns)**.** Consider

$$A = [A_1, \dots, A_n]_{m\times n}\,.$$

A permutation is a function $(1, \dots, n) \mapsto (\sigma(1), \dots, \sigma(n))$, inducing the permuted matrix A_σ

$$A_\sigma = \left[A_{\sigma(1)}, \dots, A_{\sigma(n)}\right].$$

With the same permutation for x, we have

$$x_\sigma = \begin{pmatrix} x_{\sigma(1)} \\ \vdots \\ x_{\sigma(n)} \end{pmatrix}.$$

We then easily see that

$$Ax = \sum_{j=1}^{n} A_i x_i = \sum_{j=1}^{n} A_{\sigma(j)} x_{\sigma(j)}.$$

Hence,

$$Ax = b \Leftrightarrow A_\sigma x_\sigma = b.$$

2.2 Basic Feasible Solutions and Extreme Points

2.2.1 Basic Partition

Let's first define the so-called partition, where the intention will become clear soon.

Definition 2.2.1 (Partition). A *partition* (β, η) of $\{1, \dots, n\}$ is defined as

$$\beta \coloneqq (\beta_1, \dots, \beta_m), \quad \eta \coloneqq (\eta_1, \dots, \eta_{n-m}),$$

Definition 2.2.2 (Basis). β is called *basis*.

Definition 2.2.3 (Non-basis). η is called *non-basis*.

The main idea of introducing partition is because we want to characterize what subset of the structured constraints is solvable, i.e., some submatrix A' of A is invertible. Definition 2.2.1 induces the following.

Definition 2.2.4 (Basic partition). A partition is a *basic partition* if

$$A_\beta = [A_{\beta_1}, \dots, A_{\beta_m}]_{m\times m}$$

is invertible.

2.2.2 Basic Feasible Solutions

With the notion of basic partition, we define the following.

Definition 2.2.5 (Basic solution). The *basic solution* $\overline{x}$ for a basic partition is defined as

$$\overline{x}_\eta = \begin{pmatrix} \overline{x}_{\eta_1} \\ \vdots \\ \overline{x}_{\eta_{n-m}} \end{pmatrix} \coloneqq \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \overline{x}_\beta = \begin{pmatrix} \overline{x}_{\beta_1} \\ \vdots \\ \overline{x}_{\beta_m} \end{pmatrix} \coloneqq A_\beta^{-1} b.$$

Intuition. This of course makes sense, since we know that if this is a feasible solution for a standard form problem, then $A\overline{x} = b$, which means

$$[A_\beta, A_\eta] \begin{pmatrix} \overline{x}_\beta \\ \overline{x}_\eta \end{pmatrix} = b \Rightarrow A_\beta \overline{x}_\beta + A_\eta \underbrace{\overline{x}_\eta}_{=0} = b \Rightarrow \overline{x}_\beta = \underbrace{A_\beta^{-1}}_{\text{invertible}} b$$

Remark. After choosing η, we see that $\overline{x}_\beta$ is determined.

Lecture 5: Convex Set and Extreme Points

2.2.3 Convex Sets 15 Sep. 08:00

We now turn our focus to the geometry of the linear program. Let's first study perhaps one of the most important class of sets, the convex sets.

Definition 2.2.6 (Convex set). A set $S \subseteq \mathbb{R}^n$ is *convex* if for any $x^1, x^2 \in S$ and $0 < \lambda < 1$,

$$\lambda x^1 + (1-\lambda) x^2 \in S.$$

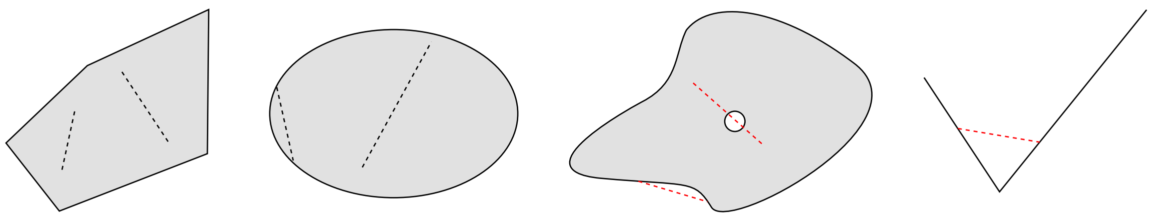


Figure 2.1: The first two are convex sets, while the latter two are not since (some parts of) those red lines are outside the set.

Intuition. A convex set is a set that contains every line segment between two points in which.

Remark. The feasible region S of any linear program is a convex set.

Proof. Consider a standard form problem, and suppose there are two feasible points x^1 and $x^2 \in S$. Then

$$\begin{cases} Ax^1 = b, & x^1 \geq 0; \\ Ax^2 = b, & x^2 \geq 0. \end{cases}$$

This implies

$$A\underbrace{\left(\lambda x^1 + (1-\lambda)x^2\right)}_{\geq 0} = \lambda Ax^1 + (1-\lambda)Ax^2 = (\lambda + (1-\lambda))b = b$$

for every $\lambda \in (0,1)$. With the fact that $\lambda x^1 + (1-\lambda)x^2$ is non-negative, hence it's feasible. ⊛

2.2.4 Extreme Points

Now, the importance of convex sets is illustrated via the following notion.

Definition 2.2.7 (Extreme point)**.** Suppose S is a convex set, then $\hat{x} \in S$ is an *extreme point* of S if we **cannot** write

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

with $x^1 \neq x^2$, $x^1, x^2 \in S$, $0 < \lambda < 1$.

Then we have an important theorem.

Theorem 2.2.1. Every basic feasible solution of standard form problem (P) is an extreme point of the feasible region of (P).

Proof. Consider a basic feasible solution $\overline{x}$: $\overline{x}_\eta = \vec{0}$, $\overline{x}_\beta = A_\beta^{-1}b \geq \vec{0}$. If it is not an extreme point, then we have

$$\exists x^1 \neq x^2 \text{ which is feasible, for } 0 < \lambda < 1 \text{ with } \overline{x} = \lambda x^1 + (1-\lambda)x^2,$$

we will have

$$\overline{x}_\eta = \underbrace{\lambda}_{>0}\underbrace{x_\eta^1}_{>0} + \underbrace{(1-\lambda)}_{>0}\underbrace{x_\eta^2}_{\geq 0} \Rightarrow x_\eta^1 = x_\eta^2 = 0 \Rightarrow x_\beta^1 = x_\beta^2 = A_\beta^{-1}b.$$

Hence, we see that $\overline{x} = x^1 = x^2$ ↯ ∎

The converse is also true, but it's harder to show.

Theorem 2.2.2. If $\hat{x}$ is an extreme point of the feasible region of (P), then $\hat{x}$ is basic.

The proof of this Theorem 2.2.2 is left as an exercise.

Lecture 6: Feasible Direction and Ray

2.3 Basic Feasible Rays and Extreme Rays

20 Sep. 08:00

2.3.1 Basic Directions

Let's first play around with the standard form problem a bit. Consider

$$\begin{aligned} \min\ & c^\top x \\ & Ax = b \\ & x \geq 0. \end{aligned}$$

It's obvious that it's equivalent to

$$\begin{aligned} \min\ & c_\beta^\top x_\beta + c_\eta^\top x_\eta \\ & A_\beta x_\beta + A_\eta x_\eta = b \\ & x_\beta \geq 0, x_\eta \geq 0. \end{aligned}$$

Further, we have

$$\begin{aligned} \min\ & c_\beta^\top (A_\beta^{-1} b - A_\beta^{-1} A_\eta x_\eta) + c_\eta^\top x_\eta \\ & x_\beta + A_\beta^{-1} A_\eta x_\eta = A_\beta^{-1} b \\ & x_\beta \geq 0, x_\eta \geq 0 \end{aligned}$$

since from the constraints, we have $x_\beta = A_\beta^{-1} b - A_\beta^{-1} A_\eta x_\eta$. Observe that the objective function now only depends on x_η, hence

$$\begin{aligned} c_\beta^\top A_\beta^{-1} b + \min\ & (c_\eta^\top - c_\beta^\top A_\beta^{-1} A_\eta) x_\eta \\ & A_\beta^{-1} A_\eta x_\eta \leq A_\beta^{-1} b \\ & x_\beta \geq 0, x_\eta \geq 0. \end{aligned}$$

Note (Reduced cost). $c_\eta^\top - c_\beta^\top A_\beta^{-1} A_\eta$ is what we called *reduced costs*.

Intuitively, we want reduced costs to be zero. With this intuition, we have the following definition.

Definition 2.3.1 (Feasible direction). Let S be a convex set and suppose $\hat{x} \in S$. Then $\hat{z}$ is a *feasible direction* relative to $\hat{x}$ if there exists some $\epsilon > 0$ such that

$$\hat{x} + \epsilon \hat{z} \in S.$$

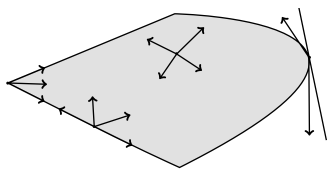


Figure 2.2: The feasible directions of a set.

Remark. For a primal (P), we must have $A\hat{z} = 0$ if $\hat{z}$ is a feasible direction.

Proof. In order to let $\hat{z}$ to be a feasible direction, we must have

$$A(\hat{x} + \epsilon \hat{z}) = \underbrace{A\hat{x}}_{=b} + \epsilon A\hat{z} = b \Leftrightarrow A\hat{z} = 0$$

⊛

Let the basic partition β, η be

$$\beta = (\beta_1, \ldots, \beta_m), \qquad \eta = (\eta_1, \underset{\underset{\eta_j}{\uparrow}}{\ldots}, \eta_{n-m}),$$

where we choose j from $1 \leq j \leq n-m$, which means we choose an η_j from η. Then, we see that there is a basic direction $\overline{z}$ associated with this particular basis β and this j defined as follows.

Definition 2.3.2 (Basic direction). Given a basic partition β, η, we say that $\overline{z}$ is a *basic direction* associated with this basis β and a j such that $1 \leq j \leq n-m$ if

$$\overline{z}_{\eta_j} = 1 \Rightarrow \begin{cases} \overline{z}_\eta & := e_j \\ \overline{z}_\beta & := -A_\beta^{-1} A_{\eta_j}. \end{cases}$$

Notation. e_j is defined to be $(0, 0, \ldots, 1, 0, \ldots, 0)$, where 1 is at the j^{th} entry.

Lemma 2.3.1. Given a basic direction $\overline{z}$, $\overline{z}$ is feasible from $\overline{x}$ if

$$0 < \min \left\{ \frac{\overline{b}_i}{\overline{a}_{i\eta_j}} \geq 0 \text{ for } i \text{ such that } \overline{a}_{i\eta_j} > 0 \right\}.$$

Proof. For a basic direction $\overline{z}$ being feasible, we need to check

(a) $A(\overline{x} + \epsilon\overline{z}) = b$: Since

$$A\overline{z} = 0 \Leftrightarrow A_\beta \overline{z}_\beta + A_\eta \overline{z}_\eta = 0 \Leftrightarrow A_\beta \overline{z}_\beta + A_\eta e_j = A_\beta \overline{z}_\beta + A_{\eta_j} = 0,$$

hence $A\overline{z} = 0$ from the fact that $\overline{z}_\beta = -A_\beta^{-1} A_{\eta_j}$, which implies

$$A_\beta \overline{z}_\beta + A_{\eta_j} = A_\beta(-A_\beta^{-1} A_{\eta_j}) + A_{\eta_j} = -A_{\eta_j} + A_{\eta_j} = 0,$$

hence $A\overline{z} = 0$, which means $A(\overline{x} + \epsilon\overline{z}) = A\overline{x} = b$. $\checkmark$

(b) $\overline{x} + \epsilon\overline{z} \geq 0$: Since

$$\begin{aligned} \overline{x}_\eta + \epsilon\overline{z}_\eta &= 0 + \epsilon e_j \geq 0 \\ \overline{x}_\beta + \epsilon\overline{z}_\beta &= \underbrace{A_\beta^{-1} b}_{\geq 0} - \underbrace{\epsilon}_{>0} A_\beta^{-1} A_{\eta_j} \overset{?}{\geq} 0, \end{aligned}$$

hence we just need to make sure $\overline{x}_\beta + \epsilon\overline{z}_\beta \geq 0$. Denote $\overline{b} := A_\beta^{-1} b$, $\overline{A}_{\eta_j} := A_\beta^{-1} A_{\eta_j}$, then the requirement becomes

$$\begin{aligned} \overline{b} - \epsilon\overline{A}_{\eta_j} \geq 0 \Leftrightarrow \overline{b}_i - \epsilon\overline{a}_{i\eta_j} &\geq 0, & \text{for } i = 1, \ldots, m \\ \Leftrightarrow \underbrace{\overline{b}_i}_{\geq 0} &\geq \epsilon\overline{a}_{i\eta_j}, & \text{for } i - 1, \ldots, m. \end{aligned}$$

We finally have that for all $i = 1, \ldots, m$ such that $\overline{a}_{i\eta_j} > 0$,

$$\epsilon \leq \frac{\overline{b}_i}{\overline{a}_{i\eta_j}}.$$

Notice that if $\overline{a}_{i\eta_j} \leq 0$, there is no restriction on ϵ being ≥ 0, so the result follows.

∎

Note. Notice that we can denote A by

$$A = \begin{bmatrix} A_\eta & A_\beta \end{bmatrix}.$$

Then since A_β is invertible, so

$$A_\beta^{-1} \begin{bmatrix} A_\eta & A_\beta \end{bmatrix} = \begin{bmatrix} A_\beta^{-1} A_\eta & I \end{bmatrix}_{m\times n}.$$

Considering

$$\begin{bmatrix} I \\ -A_\beta^{-1} A_\eta \end{bmatrix},$$

we have

$$\underbrace{\begin{bmatrix} I \\ -A_\beta^{-1} A_\eta \end{bmatrix}}_{\dim(\mathrm{CS})=n-m} \underbrace{\begin{bmatrix} A_\beta^{-1} A_\eta & I \end{bmatrix}}_{\dim(\mathrm{RS})=m} = 0.$$

And since the dimension for the first matrix is $n \times (m-n)$, we see that the columns of the first matrix form a **basis** for the null space of $\begin{bmatrix} A_\eta & A_\beta \end{bmatrix}$, namely A. Furthermore, one can see that $\overline{z}$ is the j^{th} columns of $\begin{bmatrix} I \\ -A_\beta^{-1} A_\eta \end{bmatrix}$ for a choice of j.

2.3.2 Feasible Rays

Another important geometric object we're going to study is the following.

Definition 2.3.3 (Ray). Let C be a convex set. Then $\hat{z}$ is a *ray* of C of $\hat{x} \in C$ if for all $\lambda > 0$,

$$\hat{x} + \lambda \hat{z} \in C.$$

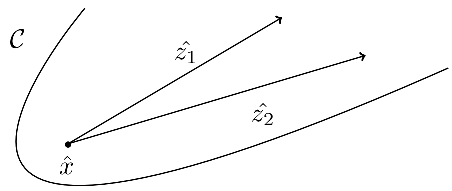


Figure 2.3: Ray of a set C.

Suppose $\hat{x} \in C$, where C is the feasible region of

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

then we see that in order to let λ arbitrarily large, we need

$$A(\hat{x} + \lambda \hat{z}) = \underbrace{A\hat{x}}_{=b} + \lambda \underbrace{A\hat{z}}_{=0} = b \Rightarrow \hat{z} \in \mathrm{NS}(A).$$

Now, observe that

$$\underbrace{\hat{x}}_{\geq 0} + \underbrace{\lambda}_{>0} \hat{z} \overset{?}{\geq} 0 \Rightarrow \hat{z} \geq 0,$$

which means that starts from the idea of basic direction, $\hat{z}$ is a ray if

$$\hat{z} \geq 0 \Leftrightarrow A_\beta^{-1} A_{\eta_j} \leq 0.$$

2.3.3 Extreme Rays

Similar to extreme point, combining this concept with ray, we have the so-called extreme ray.

Definition 2.3.4 (Extreme ray)**.** Given a convex set S, $\hat{z}$ is an *extreme ray* of S if we **cannot** write

$$\hat{z} = z^1 + z^2 \text{ with } z^1 \neq \mu z^2,$$

where z^1, z^2 being rays of S and $\mu \neq 0$.

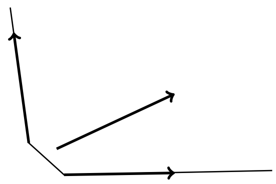


Figure 2.4: All three arrows are rays, but only the red ones are extreme.

Remark. We can compare the *non-negative basic direction* with *extreme ray*.

Basic solution $\overline{x} = \begin{cases} \overline{x}_\beta := A_\beta^{-1} b \geq 0 \\ \overline{x}_\eta := 0 \end{cases}$	$\Leftrightarrow$	Extreme points of the feasible region
basic feasible direction	v.s.	**Geometry**
(Basic direction that are non-negative)		(Extreme Ray)

Chapter 3

Simplex Algorithm

Lecture 7: Worry-Free Simplex Algorithm

3.1 A Sufficient Optimality Criterion

22 Sep. 08:00

Turns out that in order to design an algorithm which solves the primal, looking at dual is always helpful. This suggests we should first study the solution of the dual.

3.1.1 Dual Basic Solution

We start by considering the standard form problem

$$\begin{array}{lll} & \min & c^\top x \\ & & Ax = b \\ (\text{P}) & & x \geq 0, \end{array} \qquad \begin{array}{lll} & \max & y^\top b \\ & & y^\top A \leq c^\top. \\ (\text{D}) & & \end{array}$$

Definition 3.1.1 (Dual basic solution)**.** The *dual basic solution* $\overline{y} \in \mathbb{R}^m$ is defined as

$$\overline{y}^\top = c_\beta^\top A_\beta^{-1}.$$

Lemma 3.1.1. If β, η is a basic partition, and $\overline{x}$ is the associated primal basic solution and $\overline{y}$ is the associated dual basic solution, then

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

Proof.

$$c^\top \overline{x} = \begin{pmatrix} c_\beta^\top & c_\eta^\top \end{pmatrix} \begin{pmatrix} \overline{x}_\beta \\ \overline{x}_\eta \end{pmatrix} = c_\beta^\top \overline{x}_\beta + c_\eta^\top \overline{c}_\eta = c_\beta A_\beta^{-1} b = \overline{y} b.$$

■

Recall that

As previously seen.

$$\begin{aligned} \min\ & c_\beta^\top x_\beta + c_\eta^\top x_\eta \\ & A_\beta x_\beta + A_\eta x_\eta = b \\ & x_\beta \geq 0, x_\eta \geq 0, \end{aligned}$$

and hence

$$\begin{aligned} c_\beta^\top A_\beta^{-1} b + \min\ & (c_\eta^\top - c_\beta^\top A_\beta^{-1} A_\eta) x_\eta \\ & A_\beta^{-1} A_\eta x_\eta \leq A_\beta^{-1} b \\ & x_\beta \geq 0, x_\eta \geq 0. \end{aligned}$$

We now formalize the concept of reduced cost.

Definition 3.1.2 (Reduced cost)**.** The *reduced cost* $\overline{c}_\eta$ for non-basis variables is defined as

$$\overline{c}_\eta^\top \coloneqq c_\eta^\top - c_\beta^\top A_\beta^{-1} A_\eta = c_\eta^\top - \overline{y}^\top A_\eta.$$

3.1.2 Dual Feasibility

We have the following characterization of the dual basic solution $\overline{y}$.

Lemma 3.1.2. $\overline{y}$ is feasible for (D) if and only if $\overline{c}_\eta \geq 0$.

Proof. Observe that

$$y^\top A \leq c^\top \Leftrightarrow y^\top \begin{bmatrix} A_\beta & A_\eta \end{bmatrix} \leq \begin{pmatrix} c_\beta^\top & c_\eta^\top \end{pmatrix}$$

since

$$\begin{cases} y^\top A_\beta \leq c_\beta^\top \\ y^\top A_\eta \leq c_\eta^\top \Rightarrow c_\eta^\top - y^\top A_\eta \geq 0. \end{cases}$$

■

Corollary 3.1.1. Suppose $\hat{x}$ is feasible for (P) and $\hat{y}$ is feasible for (D). If $c^\top \hat{x} = \hat{y}^\top b$, then $\hat{x}$ and $\hat{y}$ are optimal.

Theorem 3.1.1 (Weak optimal basis theorem)**.** Let $\overline{x}$ and $\overline{y}$ are basic primal and dual solutions for (P) and (D). Then if β is a feasible basis and $\overline{c}_\eta \geq 0$, $\overline{x}$ and $\overline{y}$ are optimal.

Proof. Obvious from the standard problem in the form of

$$\begin{aligned} c_\beta^\top A_\beta^{-1} b + \min\ & (c_\eta^\top - c_\beta^\top A_\beta^{-1} A_\eta) x_\eta \\ & A_\beta^{-1} A_\eta x_\eta \leq A_\beta^{-1} b \\ & x_\beta \geq 0, x_\eta \geq 0. \end{aligned}$$

■

Note. The order of the arguments in text book for Theorem 3.1.1 is slightly different.

3.2 Worry-Free Simplex Algorithm

From the above discussion, we can come up with the following algorithmic approach to find the optimal solution $\overline{x}$ and $\overline{y}$ given a standard form.

(a) Start with a basis partition β, η with $\overline{x}_\beta \geq 0$.

(b) If $\overline{c}_\eta \geq 0$, then $\overline{x}$ and $\overline{y}$ are optimal, so we can stop.

(c) Otherwise, choose η_j with $\overline{c}_{\eta_j} < 0$. Consider the associated basis direction $\overline{z}$.[1] Then

$$c^\top (\overline{x} + \lambda \overline{z}) = c^\top \overline{x} + \lambda c^\top \overline{z} = c^\top \overline{x} + \lambda \overline{c}_{\eta_j},$$

where

- $c^\top \overline{x}$ is the current objective value
- $c^\top \overline{z}$ is

$$\begin{aligned} c^\top \overline{z} &= c_\eta^\top \overline{z}_\eta + c_\beta^\top \overline{z}_\beta \\ &= c_\eta^\top e_j - c_\beta^\top (A_\beta^{-1} A_{\eta_j}) \\ &= c_{\eta_j} - c_\beta^\top A_\beta^{-1} A_{\eta_j} \\ &= \overline{c}_{\eta_j} \end{aligned}$$

[1] Idea is that $\overline{x} \to \overline{x} + \lambda \overline{z}$ with $\lambda > 0$.

- $\lambda \overline{c}_{\eta_j}$ is the *rate* of change of objective value as we move in direction $\overline{z}$.

Then we move from $\overline{x}$ to $\overline{x} + \overline{\lambda}\overline{z}$, where we let $\overline{\lambda}$ as large as possible. Operationally, since we need

$$\overline{x}_\beta + \lambda \overline{z}_\beta \geq 0,$$

where $\overline{z}_\eta = e_j$, $\overline{z}_\beta = -A_\beta^{-1} A_{\eta_j}$. We then have

$$\begin{aligned} \overline{x}_{\beta_i} - \lambda \overline{a}_{i,\eta_j} &\geq 0, \text{ for } i = 1, \ldots, m \\ \lambda &\leq \frac{\overline{x}_{\beta_i}}{\overline{a}_{i,\eta_j}}, \qquad \text{for } i \text{ such that } \overline{a}_{i,\eta_j} > 0. \end{aligned}$$

Hence,

$$\overline{\lambda} := \min_{i : \overline{a}_{i,\eta_j} > 0} \left\{ \frac{\overline{x}_{\beta_i}}{\overline{a}_{i,\eta_j}} \right\} \geq 0.$$

> **Remark.** If $\overline{a}_{i,\eta_j} \leq 0$ for all $i = 1, \ldots, m$, namely
>
> $$\overline{A}_{i,\eta_j} \leq 0 \Leftrightarrow -A_\beta^{-1} A_{\eta_j} \geq 0 \Leftrightarrow \overline{z} \geq 0,$$
>
> then $\overline{z}$ is a ray. This means (P) is unbounded below, hence we terminate.

By formalizing the above procedure, we get the very first algorithmic approach to solve a linear program called the worry-free simplex algorithm. Specifically, consider the standard form problem

$$\begin{aligned} \min \ & c^\top x \\ & Ax = b \\ (\mathrm{P}) \quad & x \geq 0. \end{aligned}$$

Algorithm 3.1: Worry-Free Simplex Algorithm

Data: A standard form (P), basic partition β, η with $x_\beta \geq 0$
Result: Optimal solutions $\overline{x}, \overline{y}$, or report (P) is unbounded

while *True* **do**
  $\overline{x}_\beta \leftarrow A_\beta^{-1} b (\geq 0)$
  $\overline{c}_\eta^\top \leftarrow c_\eta^\top - c_\beta^\top A_\beta^{-1} A_\eta$
  if $\overline{c}_\eta \geq 0$ **then** // $\overline{x}$ is optimal for (P)
    $\overline{y} \leftarrow c_\beta^\top A_\beta^{-1}$ // From Theorem 3.1.1
    return $\overline{x}, \overline{y}$
  else // Basic direction $\overline{z}$, then $c^\top \overline{z} = \overline{c}_{\eta_j} < 0$
    choose j where $1 \leq j \leq n - m$ such that $\overline{c}_{\eta_j} < 0$
    if $\overline{A}_{\eta_j} \leq 0$ **then** // $\overline{A}_{\eta_j} \leq 0 \Rightarrow$ (P) is unbounded
      return (P) *is unbounded*
    $\lambda \leftarrow \min_{i:\, \overline{a}_{i\eta_j} > 0} \left\{ \frac{\overline{x}_{\beta_i}}{\overline{a}_{i,\eta_j}} \right\}$ // Largest choice so that $\overline{x} + \lambda \overline{z} \geq 0$
    $\overline{x} \leftarrow \overline{x} + \lambda \overline{z}$
    Redetermine β, η

> **Remark.** Note that x is assumed to be a basic feasible solution.

> **Problem.** The problem is that is $\overline{x} + \lambda \overline{z}$ still a basic solution? And if it is, what is the basic partition that goes with it?
>
> **Answer.** We see that after one iteration, one of the basis index i^* will become non-basis, namely
>
> $$(\overline{x} + \lambda \overline{z})_{\beta_{i^*}} = 0;$$

while one of the non-basis index will need to become basis, since

$$(\overline{x}+\lambda\overline{z})_{\beta_{i^*}} = \lambda\overline{e}_j.$$

Namely,

		$\overline{x}$	$\overline{z}$	$\overline{x}+\lambda\overline{z}$	
$\beta_{i^*} \rightarrow$	β	$\overline{x}_\beta$	$\overline{z}_\beta$	$\rightarrow 0$	β_{i^*} *becomes non-basis*
	η	$\overline{x}_\eta = 0$	$\overline{x}_\eta = e_j$	$\lambda\overline{e}_j$	η_j *becomes basis*

⊛

Now, suppose i^* is that chosen index with $\overline{a}_{i^*\eta_j} > 0$ and $\frac{\overline{x}_{\beta_{i^*}}}{\overline{a}_{i^*\eta_j}} = \overline{\lambda}$. Then we have β_{i^*} such that

$$\overline{x}+\lambda\overline{z} \Rightarrow \overline{x}_{\beta_{i^*}} + \overline{\lambda}\overline{z}_{\beta_{i^*}} = \overline{x}_{\beta_{i^*}} + \frac{\overline{x}_{\beta_{i^*}}}{\overline{a}_{i^*\eta_j}}\left(-\overline{a}_{i^*,\eta_j}\right) = 0.$$

So we reasonably suspect that there is a new basic partition such that

$$\begin{gathered}\widetilde{\beta} \coloneqq (\beta_1, \beta_2, \dots, \beta_{i^*-1}, \eta_j, \beta_{i^*+1}, \dots, \beta_m) \\ \updownarrow \\ \widetilde{\eta} \coloneqq (\eta_1, \eta_2, \dots, \eta_{j-1},\ \beta_{i^*}, \eta_{j+1}, \dots, \eta_{n-m}).\end{gathered}$$

The remaining question is that, is $A_{\widetilde{\beta}}$ still invertible? Namely, is $\det A_{\widetilde{\beta}} \neq 0$?

Lemma 3.2.1. After one iteration of worry-free simplex algorithm, $A_{\widetilde{\beta}}$ is still invertible.

Proof. We see that $A_{\widetilde{\beta}}$ is invertible if and only if $A_\beta^{-1}A_{\widetilde{\beta}}$ is invertible. And since

$$A_\beta^{-1}A_{\widetilde{\beta}} = \begin{bmatrix} e_1 & e_2 & \dots & e_{i^*-1} & \overline{A}_{\eta_j} & e_{i^*+1} & \dots & e_m \end{bmatrix},$$

and since $\det\left(A_\beta^{-1}A_{\widetilde{\beta}}\right) = \overline{a}_{i^*\eta_j}$, if $\overline{a}_{i^*\eta_j} \neq 0$, it is indeed invertible. But this is an obvious fact by our choice of i^*. ■

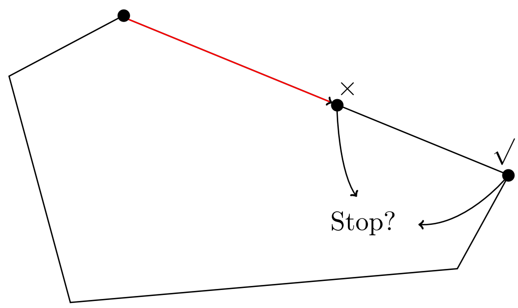


Figure 3.1: Pivot swap in terms of feasible region.

Finally, we check that the unique basic solution for this basic partition $\widetilde{\beta}$, $\widetilde{\eta}$ are exactly $\overline{x}+\overline{\lambda}\overline{z}$.

Lemma 3.2.2. The unique solution of $Ax = b$ having $x_{\widetilde{\eta}} = 0$ is $\overline{x}+\overline{\lambda}\overline{z}$.

Proof. Firstly, $(\overline{x}+\overline{\lambda}\overline{z})_j = 0$ for $j \in \widetilde{\eta}$. Moreover, $\overline{x}+\overline{\lambda}\overline{z}$ is the unique solution to $Ax = b$ having

$x_{\tilde{\eta}} = 0$ because $A_{\tilde{\beta}}$ is invertible, namely

$$Ax = b \Rightarrow \underbrace{\cancel{A_{\tilde{\eta}} x_{\tilde{\eta}}}}_{=0} + A_{\tilde{\beta}} x_{\tilde{\beta}} = b \Rightarrow x_{\tilde{\beta}} = A_{\tilde{\beta}}^{-1} b.$$

■

Lecture 8: Simplex Algorithm

3.3 Remaining Problems

27 Sep. 08:00

Now, a big question is, how do we start with a basic feasible partition? The answer is to consider the so-called phase one problem.

3.3.1 Phase one problem

Consider the following problem.

Definition 3.3.1 (Phase one problem). Given the primal (P), the *phase one problem* (Φ) is defined as follows.

$$\begin{array}{llll} & \min\ c^\top x & & \min\ x_{n+1} \\ & Ax = b & & Ax + A_{n+1} x_{n+1} = b \\ (\mathrm{P}) & x \geq 0. & (\Phi) & x \geq 0, x_{n+1} \geq 0. \end{array}$$

Remark. We see that

- If min value of x_{n+1} in (Φ) is 0, then we get a feasible solution of (P).
- If min value of x_{n+1} in (Φ) is > 0, then there is no feasible solution of (P).

By solving (Φ), we will get a feasible solution for (P) or determine whether (P) is solvable in the first place. But to solve the linear program (Φ), we're facing the same problem as (P).

Problem 3.3.1. How do we get an initial basic feasible solution for (Φ)?

Answer. In this case, we know how to get a basic feasible solution for (Φ).

(a) Start with a basic solution of (P), $\tilde{\beta}, \tilde{\eta}$ is the basic partition.

(b) If $\overline{x}_{\tilde{\beta}}$ is feasible then we just use $\tilde{\beta}$ and $\tilde{\eta}$ for β and η.

(c) Otherwise, set $A_{n+1} = -A_{\beta}^{-1}\vec{1}$. If $\eta_j = n+1$

$$\overline{z} : \overline{z}_{\tilde{\eta}} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}, \qquad \overline{z}_{\beta} := -A_{\tilde{\beta}}^{-1}(A_{n+1}) = \vec{1}$$

and

$$\vec{x} \to \vec{x} + \lambda\vec{z} \geq \vec{0}.$$

Example.

$$\vec{x}_{\tilde{\beta}} + \lambda \vec{z}_{\tilde{\beta}} = \begin{pmatrix} 7 \\ 0 \\ 3 \\ -5 \\ 6 \\ -8 \end{pmatrix} + \lambda \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix},$$

then

$$i^* = \underset{i : \vec{x}_{\tilde{\beta}} < 0}{\arg\min} \{-\vec{x}_{\tilde{\beta}}\}.$$

⊛

Remark. If $x_{n+1} = 0$, we can just stop right before $x_{n+1} = 0$, let other variable do that.

3.3.2 Perturbed Problem

Though we now know how to get basic feasible solution for (P) to start our worry-free simplex algorithm, we have one more problem.

Problem 3.3.2 (Degenerate problem)**.** What if $\lambda = 0$, i.e. $x_{\beta_i} = 0$ for some i?

In this case, we'll need the so-called non-degeneracy hypothesis.

Conjecture 3.3.1 (Non-degeneracy hypothesis)**.** At every iteration of worry-free simplex algorithm, $x_{\beta_i} > 0$ for all i.

Remark (Termination analysis)**.** With non-degeneracy hypothesis, we see that worry-free simplex algorithm will certainly terminate.

Proof. We have

$$\begin{aligned} \vec{x}_{\beta_i} > 0 \text{ for all } i \text{ at every iteration} \Rightarrow & \overline{\lambda} \neq 0 \\ \Rightarrow & \text{ objective value decrease at each iteration.} \\ \Rightarrow & \text{ Algorithm 3.1 must terminate} \end{aligned}$$

because there are only finitely many bases. ⊛

Though the non-degeneracy hypothesis helps avoid the mess, but since we want to be able to solve any general linear program, hence we now try to avoid using this hypothesis. We first consider the following problem called perturbed problem.

Definition 3.3.2 (Perturbed problem)**.** Given a standard form problem, the following induced problem is called *perturbed problem.*

$$\begin{aligned} \min\ & c^\top x \\ & Ax = b + B \begin{pmatrix} \epsilon \\ \epsilon^2 \\ \epsilon^3 \\ \vdots \\ \epsilon^m \end{pmatrix} \\ & x \geq 0 \end{aligned}$$

where ϵ is an arbitrarily small *indeterminate*.

Remark. Note that $\epsilon \neq 0$.

Note. We see that

$$\vec{x}_\beta = A_\beta^{-1}\left(b + B\begin{pmatrix}\epsilon\\ \epsilon^2\\ \vdots\\ \epsilon^m\end{pmatrix}\right) = A_\beta^{-1}b + A_\beta^{-1}B\begin{pmatrix}\epsilon\\ \epsilon^2\\ \vdots\\ \epsilon^m\end{pmatrix},$$

which is just a polynomial in ϵ.

Definition 3.3.3 (Polynomial in ϵ)**.** We denote polynomials in ϵ as

$$p(\epsilon) = p_0 + p_1\epsilon + p_2\epsilon^2 + \cdots + p_m\epsilon^m,$$

where $p_i \in \mathbb{R}$.

Which suggest the following definitions.

Definition 3.3.4 (Sign of polynomial in ϵ)**.** Let K be the minimal index with $p_K \neq 0$.

- If $p_K < 0$, then $p(\epsilon) < 0$
- If $p_K > 0$, then $p(\epsilon) > 0$
- If $p_K = 0$, namely $p_0 = p_1 = \cdots = p_m = 0$, then $p(\epsilon) = 0$

Note. Given

$$\begin{aligned} p(\epsilon) &= p_0 + p_1\epsilon + p_2\epsilon^2 + \cdots + p_m\epsilon^m \\ q(\epsilon) &= q_0 + q_1\epsilon + q_2\epsilon^2 + \cdots + q_m\epsilon^m \end{aligned}$$

with K_p and K_q. Then K_{p+q} depends on K_p and K_q. We then see that if $p(\epsilon) - q(\epsilon) \geq 0$, then $p(\epsilon) \geq q(\epsilon)$.

Problem. Where does this ϵ thing links with the worry-free simplex algorithm, and how can it solve the degenerate problem?

Answer. Suppose

$$\overbrace{p(\epsilon)}^{\substack{\text{value of}\\ \text{some basic}\\ \text{variable}}} = p_0 + p_1\epsilon + p_2\epsilon^2 + \cdots + p_m\epsilon^m.$$

Feasibility for the perturbed problem means $p(\epsilon) \geq \vec{0} \Rightarrow p(0) = p_0 \geq 0$.

$$\begin{aligned} \min\ & c^\top x \\ & Ax = b + \cancel{B}\vec{\epsilon} \\ & x \geq 0. \end{aligned}$$

Find an initial feasible basis β, η for unperturbed problem, $B := A_\beta$,

$$\vec{x}_\beta = A_\beta^{-1}(b + A_\beta\vec{\epsilon}) = \underbrace{A_\beta^{-1}b}_{\geq\vec{0}} + \vec{\epsilon} = \vec{x}_\beta + \begin{pmatrix}\epsilon\\ \epsilon^2\\ \epsilon^3\\ \vdots\\ \epsilon^m\end{pmatrix} = \begin{pmatrix}\vec{x}_{\beta_1} + \epsilon\\ \vec{x}_{\beta_2} + \epsilon^2\\ \vdots\\ \vec{x}_{\beta_m} + \epsilon^m\end{pmatrix} \geq \vec{0}.$$

Claim. Perturbed problem is non-degenerate.

Proof. This is equivalent to show that there are no i in the later basis $\tilde{\beta}$ such that $\vec{x}_{\tilde{\beta}_i} = 0$. Suppose there is an i such that $\vec{x}_{\tilde{\beta}_i} = 0$. But since

$$\vec{x}_{\tilde{\beta}} := A_{\tilde{\beta}}^{-1}(b + A_\beta \vec{\epsilon}) = A_{\tilde{\beta}}^{-1} b + A_{\tilde{\beta}}^{-1} A_\beta \vec{\epsilon},$$

if $\vec{x}_{\tilde{\beta}_i} = 0$, we must have

$$\begin{aligned} i^{\text{th}} \text{ element of } A_{\tilde{\beta}}^{-1} A_\beta \begin{pmatrix} \epsilon \\ \epsilon^2 \\ \epsilon^3 \\ \vdots \\ \epsilon^m \end{pmatrix} &= 0 \Rightarrow \left\langle i^{\text{th}} \text{ row of } A_{\tilde{\beta}}^{-1} A_\beta,\ \vec{\epsilon} \right\rangle = 0 \\ &\Rightarrow i^{\text{th}} \text{ row of } A_{\tilde{\beta}}^{-1} A_\beta = \vec{0} \ \lightning \end{aligned}$$

because $A_{\tilde{\beta}}^{-1} A_\beta$ is invertible where $A_\beta^{-1} A_{\tilde{\beta}}$ is its inverse. ⊛

⊛

Lecture 9: Practical Simplex Algorithm

29 Sep. 08:00

Note (A_β^{-1} in reality). In reality, we don't really calculate A_β^{-1}, since when calculating

$$A_\beta x_\beta = b,$$

we do not use $\overline{x}_\beta = A_\beta^{-1} b$, instead, we use LU-Factorization. And since after applying pivot change, there is only a column change in A_β^{-1}, we can use the previous result to calculate the new $\overline{x}_\beta$ much faster.

3.4 The Word *Simplex*

For a standard form problem

$$\begin{aligned} \min\ & c^\top x \\ & Ax = b \\ \text{(P)}\quad & x \geq 0, \end{aligned}$$

we can instead consider an equivalent problem formulated as

$$\begin{aligned} \min\ & z \\ & z - c^\top x = 0 \Leftrightarrow (c^\top x = z) \\ & Ax = b \\ & x \geq 0. \end{aligned}$$

As previously seen. Our picture is in $\mathbb{R}^{n-m}$, but we consider *Dantzig picture*, which is in $\mathbb{R}^{m+1}$

3.4.1 Column geometry

Plot columns:

$$\underbrace{\begin{pmatrix} c_1 \\ A_1 \end{pmatrix} \begin{pmatrix} c_2 \\ A_2 \end{pmatrix} \cdots \begin{pmatrix} c_n \\ A_n \end{pmatrix}}_{n \text{ points in } \mathbb{R}^{m+1}}$$

The requirement line is

$$\begin{pmatrix} z \\ b \end{pmatrix}.$$

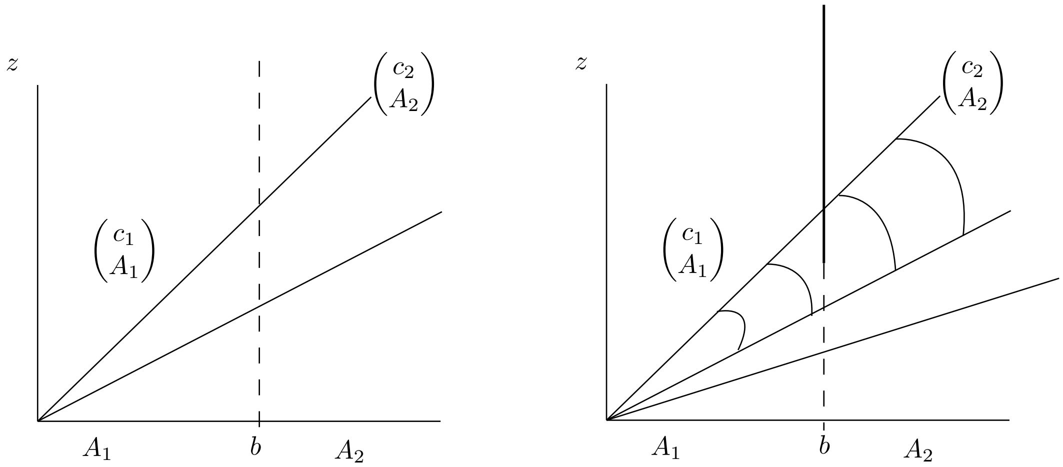


Figure 3.2: Column Geometry.

3.4.2 Simplices (Plural of Simplex)

Example (Simplex)**.** An $n-1$ dimensional simplex in $\mathbb{R}^n$ with n standard unit vectors are the corner can be described as

$$\left\{ x \in \mathbb{R}^n : \sum_{i=1}^{n} x_i = 1, x_i \geq 0 \right\}.$$

Figure 3.3: Simplex.

Note. $m+1$ points of a simplex of dimension.

A simplicial cone is rather simple, the graph below is informative enough.

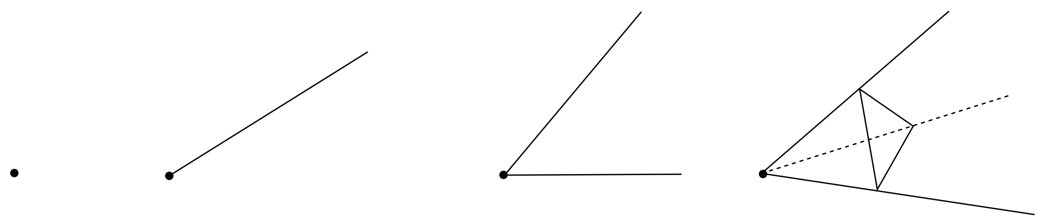


Figure 3.4: Simplicial Cones

3.5 The Simplex Algorithm

Now, given a standard form problem (P):

$$
\begin{aligned}
& \min\ c^\top x \\
& \quad\ Ax = b \\
(\mathrm{P}) & \quad\ x \geq 0,
\end{aligned}
$$

we can then solve (P) in a mathematical rigorous and complete way.

Algorithm 3.2: Simplex Algorithm

Data: standard form LP (P)
Result: optimal solutions $\overline{x}, \overline{y}$, or report (P) is unbounded/infeasible

```
(Φ_ε) ← algebraic perturbation to the phase one problem (Φ)
β ← basic feasible solution for (Φ_ε)
result ← WorryFreeSimplexAlgorithm((Φ_ε), β) [a]
if result is unbounded then
    return (P) has no solution
else
    β ← result                                // Retrieve the feasible basis for (P)

(P_ε) ← algebraic perturbation to (P)
return WorryFreeSimplexAlgorithm((P_ε), β) [b]
```

[a] Adapted to algebraically perturbed problems, and always giving preference to x_{n+1} for leaving the basis whenever it's eligible to leave for the unperturbed problem.
[b] Again, adapted to algebraically perturbed problems.

Chapter 4

Duality

Consider the standard problem and its dual

$$\begin{array}{llll} & \min\ c^\top x & & \max\ y^\top b \\ & Ax = b & & y^\top A \leq c^\top. \\ \text{(P)} & x \geq 0, & \text{(D)} & \end{array}$$

4.1 The Strong Duality Theorem

As previously seen. The weak duality theorem states that if $\hat{x}$ is feasible for (P), and $\hat{y}$ is feasible for (D), then

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

Moreover, the equality holds if and only if $\hat{x}$ and $\hat{y}$ are optimal.

As previously seen. The weak optimal basis theorem states that if we have a basic partition β, η, and we also have $\overline{x}_\beta \geq \vec{0}$($\overline{x}$ is feasible for (P)) and $\overline{c}_\eta \geq \vec{0}$($\overline{y}$ is feasible for (D)), then $\overline{x}$ and $\overline{y}$ are both optimal.

Now, we have so-called strong optimal basis theorem.

Theorem 4.1.1 (Strong optimal basis theorem)**.** If (P) has a feasible solution, and if (P) is not unbounded, then there exist a basic partition β, η such that $\overline{x}$ and $\overline{y}$ are optimal, and

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

Proof. Since if (P) has a feasible solution and is not unbounded, we can just run the simplex algorithm, which will terminate with a basis β such that the associated basic solution $\overline{x}$ and the associated dual solution $\overline{y}$ are optimal. ∎

We see that this leads to another similar result.

Theorem 4.1.2 (Strong duality theorem)**.** If (P) has a feasible solution and (P) is not unbounded, then there exist optimal solutions $\hat{x}$ and $\hat{y}$ with

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

Note. The proof of these two theorems are by directly using the *mathematical complete* version of simplex algorithm, hence the completeness of simplex algorithm (namely the phase one problem and the perturbation) is important.

Simplex Algorithm	(P)\(D)	optimal solution	infeasible	unbounded
$\overline{c}_\eta \geq \vec{0} \Rightarrow$ Stop	optimal solution	$\surd$	$\times$	$\times$
optimal x_{n+1} in Φ is positive	infeasible	$\times$	$\surd$	$\surd$
$\overline{A}_{\eta_j} \leq \vec{0} \Rightarrow$ Stop	unbounded	$\times$	$\surd$	$\times$

Table 4.1: Comparison between (P) and (D)

Lecture 10: Complementary Slackness

4.2 Complementary Slackness

4 Oct. 08:00

Besides strong optimal basis theorem and strong duality theorem, we have even more connection between the dual and the optimality condition.

Definition 4.2.1 (Complementary). Solutions $\hat{x}$ to (P) and $\hat{y}$ to (D) are *complementary* if

$$\begin{aligned}(c_j - \hat{y}^\top A_{\cdot j})\hat{x}_j = 0, &\quad j = 1 \cdots n;\\ \hat{y}_i(A_{i\cdot}\hat{x} - b_i) = 0, &\quad i = 1 \cdots m.\end{aligned}$$

Now, suppose we have a basic partition β, η such that

$$\begin{aligned}&\overline{x} \colon \overline{x}_\beta = A_\beta^{-1} b,\ \overline{x}_\eta = \vec{0}\\ &\overline{y} \colon \overline{y}^\top = c_\beta^\top A_\beta^{-1},\end{aligned}$$

then

$$\begin{aligned}\underbrace{(c_j - \hat{y}^\top A_{\cdot j})}_{=0 \text{ for } j\in\beta} \underbrace{\hat{x}_j}_{\substack{=0\\ \text{for}\\ j\in\eta}} &= 0, j = 1 \cdots n;\\ \hat{y}_i \underbrace{(A_{i\cdot}\hat{x} - b_i)}_{=0 \text{ for } \overline{x}} &= 0, i = 1 \cdots m.\end{aligned}$$

Note. Specifically, $c_j - \hat{y}^\top A_{\cdot j} = 0$ for $j \in \beta$ since $\overline{y}^\top = c_\beta^\top A_\beta^{-1}$, and

$$c_j - \hat{y}^\top A_{\cdot j} = c_j - c_\beta^\top \underbrace{A_\beta^{-1} A_{\cdot j}}_{e_j} = c_j - c_j = 0.$$

Then just from above, we see that the following theorems hold.

Theorem 4.2.1. If $\overline{x}$ and $\overline{y}$ are basic solutions[a] for β, η, then $\overline{x}$ and $\overline{y}$ are complementary.

[a] Can be either feasible or not.

Theorem 4.2.2. If $\hat{x}$ and $\hat{y}$ are complementary with respect to (P) and (D), then $c^\top \hat{x} = \hat{y}^\top b$.

Note.

$$c_\beta^\top A_\beta^{-1} b = \overline{y}^\top b, \qquad c^\top (A_\beta^{-1} b) = c_\beta^\top \overline{x}_\beta = c^\top \overline{x}.$$

Proof. We show that

$$c^\top \hat{x} - \hat{y}^\top b = 0.$$

We have

$$\begin{aligned} c^\top \hat{x} - \hat{y}^\top b &= (c^\top \underbrace{- \hat{y}^\top A)\hat{x} + \hat{y}^\top (A\hat{x}}_{\text{added terms}} - b) \\ &= \sum_{j=1}^{n} \underbrace{(c_j - \hat{y}^\top A_{\cdot j})x_j}_{=0 \text{ for } j=1\dots n} + \sum_{i=1}^{m} \underbrace{\hat{y}_i(A_{i\cdot}\hat{x} - b_i)}_{=0 \text{ for } i=1\dots m} = 0. \end{aligned}$$

⊛

Theorem 4.2.3 (Weak complementary slackness theorem)**.** If $\hat{x}$ and $\hat{y}$ are feasible and complementary, then they are optimal.

Proof. Follows from Theorem 1.2.1 and complementary solutions having equal objective value from Theorem 4.2.2. ∎

Theorem 4.2.4 (Strong complementary slackness theorem)**.** If $\hat{x}$ and $\hat{y}$ are optimal, then $\hat{x}$ and $\hat{y}$ are complementary.

Proof. Recall that

$$\overbrace{\sum_{j=1}^{n} \underbrace{(c_j - \hat{y}^\top A_{\cdot j})}_{\geq 0 \text{ for each } j} \quad \underbrace{\hat{x}_j}_{\geq 0 \text{ for each } j}}^{\cancel{\geq 0} \Rightarrow = 0 \text{ for each } j} + \overbrace{\sum_{i=1}^{m} \underbrace{\hat{y}_i(A_{i\cdot}\hat{x} - b_i)}_{=0 \text{ for each } i}}^{=0} \underbrace{= 0 = c^\top \hat{x} - \hat{y}^\top b}_{\substack{\text{if } \hat{x} \text{ and } \hat{y} \text{ are optimal:} \\ \text{same object value}}}$$

Hence, the equality can only hold if

$$(c_j - \hat{y}^\top A_{\cdot j})\hat{x}_j = 0, \text{ for } j = 1, 2, \dots, n;$$

with the obvious fact that

$$\hat{y}_i(A_{i\cdot}\hat{x} - b_i) = 0, \text{ for } i = 1, 2, \dots, m,$$

so they are complementary. ∎

4.3 Duality for General Linear Optimization Problems

So far, we only discuss the dual of the standard form problem. But we will see that *every* linear optimization problem has a natural dual. Now consider a general linear programming problem

$$\begin{aligned} &\min \ c_P^\top x_P + c_N^\top x_N + c_U^\top x_U \\ &\qquad A_{GP}x_P + A_{GN}x_N + A_{GU}x_U \geq b_G \\ &\qquad A_{LP}x_P + A_{LN}x_N + A_{LU}x_U \leq b_L \\ &\qquad A_{EP}x_P + A_{EN}x_N + A_{EU}x_U = b_E \\ (\mathcal{G}) &\qquad x_P \geq 0, x_N \leq 0, x_U \text{ unrestricted.} \end{aligned}$$

We first turn this into a standard form problem:

(a) $\tilde{x}_N := -x_N$:

$$\begin{aligned} &\min \ c_P^\top x_P + c_N^\top x_N + c_u^\top x_U \\ &\qquad A_{GP}x_P - A_{GN}x_N + A_{GU}x_U \geq b_G \\ &\qquad A_{LP}x_P - A_{LN}x_N + A_{LU}x_U \leq b_L \\ &\qquad A_{EP}x_P - A_{EN}x_N + A_{EU}x_U = b_E \\ &\qquad x_P \geq 0, x_N \leq 0, x_U \text{ unrestricted} \end{aligned}$$

(b) $x_U = \widetilde{x}_U - \widetilde{\widetilde{x}}_U$, where $\widetilde{x}_U, \widetilde{\widetilde{x}}_U \geq 0$:

$$
\begin{aligned}
\min\ & c_P^\top x_P + c_N^\top x_N + c_U^\top \widetilde{x}_U - c_U \widetilde{\widetilde{x}}_U \\
& A_{GP}x_P - A_{GN}x_N + A_{GU}\widetilde{x}_U - A_{GU}\widetilde{\widetilde{x}}_U \geq b_G \\
& A_{LP}x_P - A_{LN}x_N + A_{LU}\widetilde{x}_U - A_{LU}\widetilde{\widetilde{x}}_U \leq b_L \\
& A_{EP}x_P - A_{EN}x_N + A_{EU}\widetilde{x}_U - A_{EU}\widetilde{\widetilde{x}}_U = b_E \\
& x_P \geq 0, x_N \leq 0, \widetilde{x}_U \geq 0, \widetilde{\widetilde{x}}_U \geq 0
\end{aligned}
$$

(c) Adding slack variables:

$$
\begin{aligned}
\min\ & c_P^\top x_P + c_N^\top x_N + c_U^\top \widetilde{x}_U - c_U \widetilde{\widetilde{x}}_U \\
& A_{GP}x_P - A_{GN}x_N + A_{Gu}\widetilde{x}_U - A_{GU}\widetilde{\widetilde{x}}_U - s_G && = b_G \\
& A_{LP}x_P - A_{LN}x_N + A_{Lu}\widetilde{x}_U - A_{LU}\widetilde{\widetilde{x}}_U && +t_L = b_L \\
& A_{EP}x_P - A_{EN}x_N + A_{Eu}\widetilde{x}_U - A_{EU}\widetilde{\widetilde{x}}_U && = b_E \\
& x_P \geq 0, x_N \leq 0, \widetilde{x}_U \geq 0, \widetilde{\widetilde{x}}_U \geq 0, s_G \geq 0, t_L \geq 0
\end{aligned}
$$

With *dual variables* y_G, y_L, y_E, we have

$$
\begin{aligned}
\max\ & y_G^\top b_G \quad + y_L^\top b_L \quad + y_E^\top b_E \\
& y_G^\top A_{GP} + y_L^\top A_{LP} + y_E^\top A_{EP} \leq c_P^\top \\
& -y_G^\top A_{GN} - y_L^\top A_{LN} - y_E^\top A_{EN} \leq -c_N^\top \\
& y_G^\top A_{GU} + y_L^\top A_{LU} + y_E^\top A_{EU} \leq c_U^\top \\
& -y_G^\top A_{GU} - y_L^\top A_{LU} - y_E^\top A_{EU} \leq -c_U^\top \\
& y_G^\top \geq 0\ , y_L^\top \leq 0.
\end{aligned}
$$

We time -1 to the both sides of the second constraint, then the last two structure constraints can be reduced to a single equality, results in

$$
\begin{aligned}
\max\ & y_G^\top b_G \quad + y_L^\top b_L \quad + y_E^\top b_E \\
& y_G^\top A_{GP} + y_L^\top A_{LP} + y_E^\top A_{EP} \leq c_P^\top \\
& y_G^\top A_{GN} + y_L^\top A_{LN} + y_E^\top A_{EN} \geq c_N^\top \\
& y_G^\top A_{GU} + y_L^\top A_{LU} + y_E^\top A_{EU} = c_U^\top \\
(\mathcal{H})\quad & y_G^\top \geq 0\ , y_L^\top \leq 0.
\end{aligned}
$$

Finally, we remark that this gives us a simple result as we have already seen before.

Theorem (Duality for general LP). Consider a general linear programming problem $(\mathcal{G})$ and $(\mathcal{H})$ obtained from $(\mathcal{G})$.

> **Theorem 4.3.1** (Weak duality theorem). If $(\hat{x}_P, \hat{x}_N, \hat{x}_U)$ is feasible in $\mathcal{G}$ and the dual variables $(\hat{y}_G, \hat{y}_L, \hat{y}_E)$ is feasible in $\mathcal{H}$, then
>
> $$c_P^\top \hat{x}_P + c_N^\top \hat{x}_N + c_U^\top \hat{x}_U \geq \hat{y}_G^\top b_G + \hat{y}_L^\top b_L + \hat{y}_E^\top b_E.$$

> **Theorem 4.3.2** (Strong duality theorem). If $\mathcal{G}$ has a feasible solution, and $\mathcal{G}$ is not unbounded, then there exist feasible solutions $(\hat{x}_P, \hat{x}_N, \hat{x}_U)$ for $\mathcal{G}$ and $(\hat{y}_G, \hat{y}_L, \hat{y}_E)$ for $\mathcal{H}$ that are optimal. Moreover,
>
> $$c_P^\top \hat{x}_P + c_N^\top \hat{x}_N + c_U^\top \hat{x}_U = \hat{y}_G^\top b_G + \hat{y}_L^\top b_L + \hat{y}_E^\top b_E.$$

Remark. We can also rephrase the Theorem 4.2.3 and Theorem 4.2.4 in this setup. The proof follows the same idea, but with some more works.

Lecture 11: Duality

6 Oct. 08:00

As previously seen (The production problem). Recall the production problem, where we have

$$\begin{aligned} \max\ & c^\top x \\ & Ax \leq b \\ & x \geq \vec{0} \end{aligned}$$

- n products activities
- c_j = per-unit revenue for activity $j = 1 \dots n$
- b_i = resource endowment for resource $i = 1 \dots m$
- a_{ij} = amount of resource i consumed by activity j

Then we have the dual as

$$\begin{aligned} \min\ & y^\top b \\ & y^\top A \geq \vec{c} \\ & y \geq \vec{0} \end{aligned}$$

where

$$y^\top A_{\cdot j} \geq c_j \left(\sum_{i=1}^{m} y_i a_{ij} \right) \geq c_j.$$

Note. We have

$$\begin{array}{r|c|c|l} & \min & \max & \\ \hline \text{constraints} \left\{ \begin{array}{l} \\ \\ \\ \end{array} \right. & \begin{array}{l} \geq \\ \leq \\ = \end{array} & \begin{array}{l} \geq 0 \\ \leq 0 \\ \text{unres.} \end{array} & \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \text{variables} \\ \hline \text{variables} \left\{ \begin{array}{l} \\ \\ \\ \end{array} \right. & \begin{array}{l} \geq 0 \\ \leq 0 \\ \text{unres.} \end{array} & \begin{array}{l} \leq \\ \geq \\ = \end{array} & \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \text{constraints} \end{array}$$

for a general rule to find a primal's dual.

Come back to complementary.

$$\begin{aligned} \hat{y}^\top A_{\cdot j} - c_j \hat{x}_j &= 0 \text{ for } j = 1 \dots n \\ \hat{y}_i (b_i - A_{i \cdot} x) &= 0 \text{ for } i = 1 \dots m \end{aligned}$$

Note. For feasible solutions of (P) and (D), at most one of $\hat{y} A_{\cdot j} - c_j$ and $\hat{x}_j$ is positive for $j = 1 \dots n$; while at most one of $b_i - A_{\cdot j} \hat{x}$ and $\hat{y}_j$ is positive for $i = 1 \dots m$;

Problem 4.3.1. We are looking for a way to find out the upper bound of $c^\top x$ from the dual.

Answer. Since

$$c^\top x \underset{?}{\leq} \underbrace{y^\top A}_{\geq c^\top} \underbrace{x}_{\geq \vec{0}} \leq \underbrace{y^\top}_{\geq \vec{0}} b \Leftrightarrow \sum_{i=1}^{m} y_i \left(\sum_{j=1}^{n} a_{ij} x_j \right) \leq \sum_{i=1}^{m} y_i b_i.$$

We want

$$c^\top \leq y^\top A \Rightarrow c^\top x \leq y^\top A x$$

Now, return to the standard form problem, we have

$$\begin{array}{llll} & \min\ c^\top x & & \max\ y^\top b \\ & Ax = b & & y^\top A \leq c^\top \\ \text{(P)} & x \geq 0 & \text{(D)} & \end{array}$$

with y unrestricted. Then we have

$$c^\top x \underset{?}{\geq} \underbrace{y^\top A}_{\leq c^\top} \underbrace{x}_{\geq 0} = y^\top b$$

since

$$y^\top Ax \leq c^\top x.$$

⊛

Intuition. For a minimization problem, we are just trying to find the lower bound of the objective function's value.

Example. Consider the following linear programming problem:

$$\begin{aligned} \max\ & c^\top x + d^\top z \\ & Ax \geq b \\ & Bx - Fz = g \\ & x \leq 0, z \text{ unrestricted} \end{aligned}$$

Then the dual is (with dual variables y, w)

$$\begin{aligned} \min\ & y^\top b\ + w^\top g \\ & y^\top A + w^\top B \leq c^\top \\ & \quad\ - w^\top F = d^\top \\ & y \leq 0, w \text{ unrestricted}, \end{aligned}$$

where we just look up the table for finding the dual. Or, we can also find the dual from

$$\begin{aligned} & y^\top A + w^\top B \leq c^\top \\ & (y^\top A + w^\top B)x \geq c^\top x' \end{aligned}$$

hence

$$\begin{array}{c} \overbrace{y^\top}^{\leq 0} (Ax \geq b) \\ + w^\top (Bx - Fz = g) \\ \hline c^\top x + d^\top z \overset{\text{want}}{\leq} \underbrace{y^\top Ax + w^\top Bx - w^\top Fz}_{\underbrace{(y^\top A + w^\top B)}_{\leq c^\top} \underbrace{x}_{\leq 0} \underbrace{-(w^\top F)z}_{=d^\top}} \overset{\text{want}}{\leq} y^\top b + w^\top g \end{array}$$

Remark. Think about what if all are equal sign? (both in constraints and variables, namely unrestricted)

4.4 Theorems of the Alternative

In this section, we want to characterize when a linear program has a feasible solution by studying the duality.

4.4.1 Farkas Lemma

Let's first study a motivating lemma called Gauss' lemma.

Lemma 4.4.1 (Gauss' Lemma). Exactly one of (I) or (II) has a solution:

$$\text{(I)} \quad Ax = b, \qquad \text{(II)} \quad \begin{aligned} y^\top A &\geq 0 \\ y^\top b &\neq 0. \end{aligned}$$

Proof. This just follows from the Gauss elimination. By doing the elimination, there are two cases.

(a) The system has no solution.

(b) There is a(some) solution(s).

For second case, it's just $Ax = b$ is solvable. For the fist case, we see that after the elimination, we will have something like

$$\begin{pmatrix} \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \\ \end{pmatrix} \begin{pmatrix} \\ a \\ \\ \end{pmatrix}$$

where $a \neq 0$, which just indicates this system is unsolvable. ■

Now, let's see the Farkas lemma.

Lemma 4.4.2 (Farkas Lemma). For any data A and b, then exactly one of (I) or (II) has a solution:

$$\text{(I)} \quad \begin{aligned} Ax &= b \\ x &\geq 0, \end{aligned} \qquad \text{(II)} \quad \begin{aligned} y^\top b &> 0 \\ y^\top A &\leq 0. \end{aligned}$$

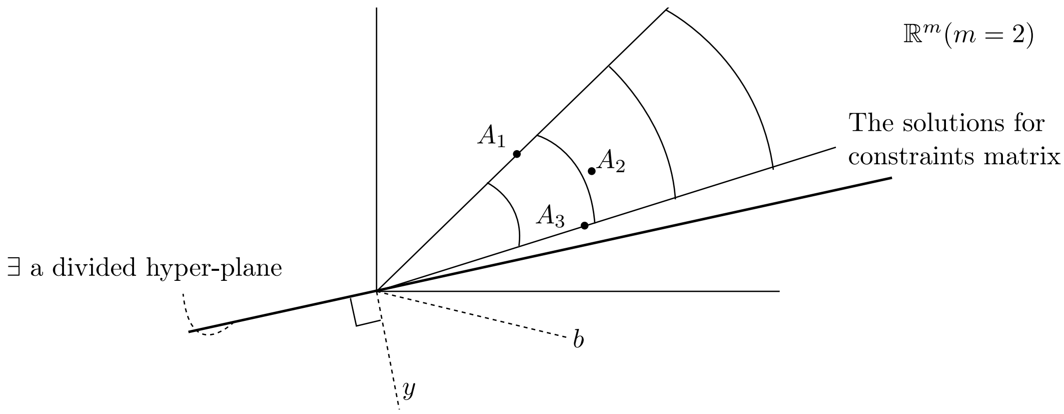


Figure 4.1: Geometrically point of view with $\mathbb{R}^m, m = 2$

Intuition. We outline the idea about the proof.

- Step 1: (I) and (II) can't both have solutions for the same A, b. Suppose $\hat{x}$ solves (I) and $\hat{y}$

solves (II). Then we have

$$0 \geq \underbrace{\hat{y}^\top A}_{\leq \vec{0}} \underbrace{\hat{x}}_{\geq \vec{0}} = \hat{y}^\top b \ \lightning$$

- Step 2: Show that if (I) has no solution, then (II) has a solution.

Lecture 12: Farkas Lemma

11 Oct. 08:00

Proof of Lemma 4.4.2. As what we have outlined, we divide the proof into two cases.

Claim. (I) and (II) can't both have solutions.

Proof. Suppose $\hat{x}$ solves I and $\hat{y}$ solves (II). Then we have

$$\hat{y}^\top (\hat{A x} = b) \Rightarrow \underbrace{(\hat{y}^\top A)}_{\geq \vec{0}} \underbrace{\hat{x}}_{\geq \vec{0}} = \hat{y} b > 0 \lightning$$

⊛

Claim. At least one of (I) or (II) has a solution ≅ If (I) has no solution, then (II) has a solution.

Proof. Assume that (I) has no solution, which means that (P) is infeasible with (P) being

$$\begin{aligned} &\min\ \vec{0}^\top x \\ &\qquad Ax = b \\ \text{(P)}\quad &\qquad x \geq 0. \end{aligned}$$

The dual of this (P) is

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

But this means that (D) is infeasible or unbounded. But we see that (D) can't be infeasible, because $y = \vec{0}$ is a feasible solution, then we know

$$\begin{aligned} &\Rightarrow D \text{ is unbounded} \\ &\Rightarrow \text{ there exist a feasible solution } \tilde{y} \text{ to (D) with positive objective.} \end{aligned}$$

⊛

■

Remark. Now, consider $\lambda\tilde{y}$ (feasible for (D)). Drive to $+\infty$ by increasing λ. We now see what Farkas Lemma really tells us.

$$\begin{array}{lll} & \min\ c^\top x & \\ & \quad Ax = b & \\ \text{(P)} & \quad x \geq 0 & \text{feasibility} \\ & & \Updownarrow \\ & \max\ y^\top b & \text{unbounded direction} \\ \text{(D)} & \quad y^\top A \leq c^\top & \end{array}$$

Suppose $\widetilde{y}$ is feasible to (D) and suppose $\hat{y}$ satisfies (II), then

$$(\widetilde{y}+\lambda\hat{y})^\top A = \underbrace{\widetilde{y}^\top A}_{\leq c^\top} + \underbrace{\lambda}_{>0} \underbrace{\hat{y}^\top A}_{\leq \vec{0}} \leq c^\top .$$

Furthermore, we have

$$(\widetilde{y}+\lambda\hat{y})^\top b = \widetilde{y}^\top b + \lambda\hat{y}^\top b \Rightarrow \infty \text{ as } \lambda\uparrow .$$

Example. Given

$$\begin{aligned} &(\mathrm{I}) \quad Ax \leq b \\ &(\mathrm{II}) \quad ?, \end{aligned}$$

find out what (II) is.

Proof. We simply set up the (P) and then find its dual.

$$\begin{array}{llll} & \min\ \vec{0}^\top x & & \max\ y^\top b \\ & Ax \leq b & & y^\top A = \vec{0}. \\ (\mathrm{P}) & & (\mathrm{D}) & y \leq \vec{0} \end{array}$$

Then we have

$$\begin{aligned} (\mathrm{I}) \quad & Ax \leq b \\ (\mathrm{II}) \quad & y^\top A = \vec{0} \\ & y \leq \vec{0} \\ & y^\top b > 0 \end{aligned}$$

Check:

$$0 = \underbrace{\hat{y}^\top A}_{=\vec{0}} \hat{x} \underset{\hat{y}\leq\vec{0}}{\geq} \hat{y}^\top b > 0 \lightning$$

or,

$$\begin{aligned} &Ax \overset{y\leq\vec{0}}{\leq} b \qquad (y^\top b > 0) \\ &0 \overset{?}{\geq} \underbrace{y^\top A}_{=\vec{0}} x \geq y^\top b > 0 \lightning \end{aligned}$$

⊛

Let's look at another example.

Example.

$$\begin{array}{ll} & (\min\ \vec{0}^\top x + \vec{0}^\top w) \\ & A\ \ x + B\ \ w = b \\ & \quad\ - F\ \ w \geq f \\ (\mathrm{I}) & \quad x \geq 0,\ w \text{ unrestricted} \end{array}$$

with the dual variables y, w, we have

$$\begin{array}{ll} & \text{(Suppose (I) has no solution.)} \\ & \cancel{\max}\ y^\top b + v^\top b (> 0) \\ & y^\top A \leq \vec{0} \\ (\mathrm{II}) & y^\top B - v^\top F = \vec{0} \end{array}$$

with y unrestricted, $v \geq \vec{0}$.

Now, we should have a general picture about what Farkas Lemma really means. For conditions (I) and (II), we have

$$\begin{array}{lll} \text{(I)} & Ax = b & \\ & x \geq 0 & \Leftrightarrow b \text{ is in the cone } K \\ \text{(II)} & y^\top b > 0 & \Leftrightarrow y \text{ makes an acute angle with } b. \\ & y^\top A \leq 0^\top & y \text{ makes a non-acute angle with all columns of } A \end{array}$$

Suppose $\hat{z}$ in K, then

$$\hat{z} = A\hat{x} \text{ for some } \hat{x} \geq \vec{0}.$$

Then we have

$$y^\top \hat{z} = \underbrace{y^\top A}_{\leq \vec{0}^\top} \underbrace{\vec{x}}_{\geq \vec{0}} \leq 0.$$

We see that y makes a non-acute angle with everything in K. Now, suppose $\hat{y}$ solves (II). Consider

$$\underbrace{\hat{y}^\top}_{\text{numbers}} \underbrace{z}_{\text{variables}} = 0.$$

Now, we have the hyperplane: $\{z \colon \hat{y}^\top z = 0\}$ separates b and K.

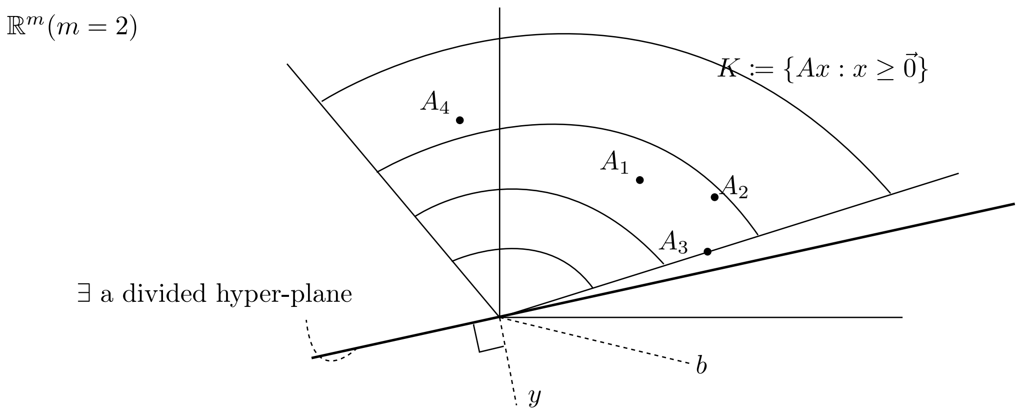


Figure 4.2: Case (II) of the Farkas Lemma with $m = 2$

4.4.2 The Big Picture of Cones

Consider the linear programming problem

$$\begin{aligned} \max \ & y^\top b \\ & y^\top A \leq c^\top \end{aligned}$$

with the partition β, η, we see that

$$y^\top A \leq c^\top \Rightarrow \begin{cases} y^\top A_\beta & \leq c_\beta^\top \\ y^\top A_\eta & \leq c_\eta^\top \end{cases}.$$

By solving only for β, then we have $\overline{y}^\top = c_\beta^\top A_\beta^{-1}$. And then, by considering the cones, we have

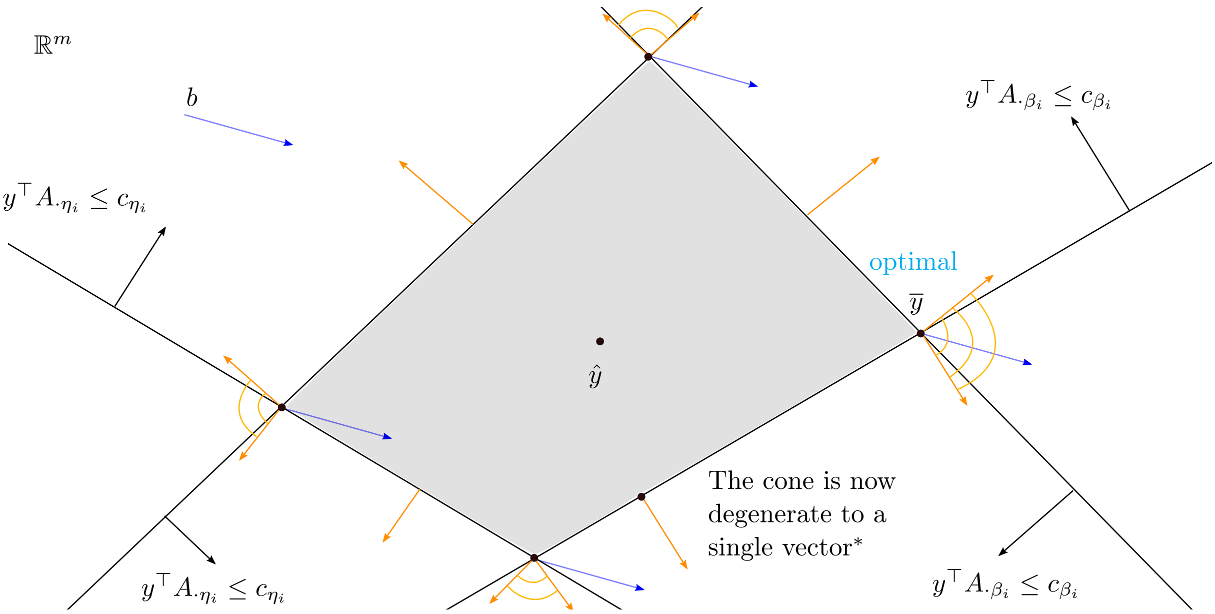


Figure 4.3: Optimality of Cones.[1]

with

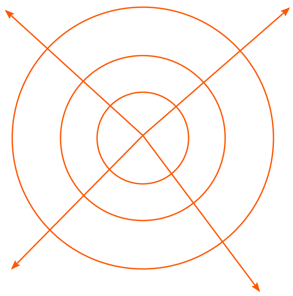


Figure 4.4: Cones join together.

Note. Consider $b = \vec{0}$ ($\hat{y}$). It's in every cone $\Rightarrow$ every point is optimal.

Remark. Each corner (extreme point) corresponds to a solution for β, while the blue vector $\vec{b}$ corresponds to the dual constraints $y^\top A_\eta < c_\eta^\top$. Only when the blue vector are in the region of orange sectors span by two *normal vectors* of $y^\top A_{\cdot \beta_i} \leq c_{\beta_i}$, the constraints are satisfied.

4.5 Strict Complementary Slackness

Consider

$$\begin{array}{llll} & \min\ c^\top x & & \max\ y^\top b \\ & \quad Ax = b & & \quad y^\top A \leq c^\top . \\ \text{(P)} & \quad x \geq 0, & \text{(D)} & \end{array}$$

[1]This corresponds to the case that we run into the overlapping issue in Figure 4.4.

As previously seen. Complementarity of $\hat{x}$ and $\hat{y}$:

$$(c_j - \hat{y}^\top A_{\cdot j})\hat{x}_j = 0, \text{ for } j = 1 \dots n$$
$$y_i^\top (A_{i\cdot}\hat{x} - b_i) = 0, \text{ for } i = 1 \dots m$$

Now, let's introduce the so-called ~~over~~ strictly complementary.

Definition 4.5.1 (Strictly complementary)**.** For feasible solutions $\hat{x}$ and $\hat{y}$ are *strictly complementary* if they are complementary and exactly one of

$$c_j - \hat{y}^\top A_{\cdot j} \text{ and } \hat{x}_j \text{ is } 0.$$

Then, we have the following [Lee22, Exercise 5.5].

Theorem 4.5.1 (Strictly complementarity)**.** If (P) and (D) are both feasible, then for (P) and (D) there exist strictly complementary (feasible) optimal solutions.

Intuition. Let v be the optimal value of (P):

$$\begin{aligned} v = \min\ & c^\top x \\ & Ax = b \\ (\mathrm{P}) \quad & x \geq 0 \end{aligned}$$

Now, we try to find an optimal solution with

$$x_j > 0, \quad \text{fix } j$$

by formulating the following linear programming

$$\begin{aligned} \max\ & x_j \\ & c^\top x \leq v \\ & Ax = b \\ (P_j) \quad & x \geq 0 \end{aligned}$$

where P_j seeks an optimal solution of (P) that has x_j being positive. If failed, then construct an optimal solution $\hat{y}$ to (D) with

$$c_j - \hat{y}^\top A_{\cdot j} > 0.$$

We then see for any **fixed** j, the desired property holds. The only thing we need to do is combine these n pairs of $\hat{x}$ and $\hat{y}$ appropriately to construct optimal $\hat{x}$ and $\hat{y}$ that are overly complementary.

Lecture 13: Duality

We now formally prove strictly complementarity theorem. 18 Oct. 08:00

Proof of Theorem 4.5.1. First prove for one fixed j. Consider

$$\begin{aligned} \max\ & x_j \\ & c^\top x \leq v \\ & Ax = b \\ (P_j) \quad & x \geq 0, \end{aligned}$$

where

$$\begin{aligned} & c^\top x \\ & Ax = b \\ & x \geq 0 \end{aligned}$$

is trying to model the set of optimal solutions to (P), and P_j is trying to find an optimal solution of (P) with $x_j > 0$.

We see that there are three cases.

1. P_j has an optimal solution. $\hat{x}$ with $\hat{x}_j > 0$. Take $\hat{x}$ optimal for $P_j \Rightarrow \hat{x}$ optimal for (P). Take a $\hat{y}$ optimal for (D).

2. P_j is unbounded. Take any feasible solutions $\hat{x}$ of P_j with $\hat{x}_j > 0$.

3. The optimal value of P_j is zero. Then consider the dual of P_j, denoted by D_j with the dual variables $w \in \mathbb{R},\ y \in \mathbb{R}^m$. We then have

$$\begin{aligned} \min \ & wv + y^\top b \\ & wc^\top + y^\top A \geq e_j^\top \\ (D_j) \quad & w \geq 0,\ y \text{ unres.} \end{aligned}$$

Suppose $\hat{w}$ and $\hat{y}$ is optimal for D_j.

Case 1. $\hat{w} > 0$: Then

$$\begin{aligned} & -c^\top + \left(\frac{\hat{y}^\top}{-\hat{w}}\right) A \underset{\leq}{\not\geq} \frac{1}{-\hat{w}} e_j^\top \\ \Rightarrow & \underbrace{\left(\frac{\hat{y}^\top}{-\hat{w}}\right)}_{\hat{\hat{y}}} A \leq c^\top - \frac{1}{\hat{w}} e_j^\top \\ \Rightarrow & \hat{\hat{y}}^\top A \leq c^\top - \frac{1}{\hat{w}_j} e_j^\top \\ \Rightarrow & \hat{\hat{y}}^\top A \leq c^\top \text{ with a little slack in the } j^{th} \text{ constraint.} \\ \Rightarrow & \hat{\hat{y}}^\top A_{\cdot j} \leq c_j - \frac{1}{\hat{w}} < c_j, \forall j. \end{aligned}$$

Note that the optimal value of D_j is zero since the optimal value of P_j is zero. Then

$$\begin{aligned} & \hat{w}v + \hat{y}^\top b = 0 \\ \Rightarrow & -v + \left(\frac{\hat{y}^\top}{-\hat{w}}\right) b = 0 \\ \Rightarrow & \hat{\hat{y}}^\top b = v \\ \Rightarrow & \hat{\hat{y}} \text{ is optimal for } D. \end{aligned}$$

Case 2. $\hat{w} = 0$: Then

$$\hat{y}^\top A \geq e_j^\top .$$

Let $\widetilde{y}$ be an optimal solution of (D). Now consider $\widetilde{y} - \hat{y}$, we have

$$(\widetilde{y} - \hat{y})^\top A = \underbrace{\widetilde{y}^\top A}_{\leq c^\top} - \underbrace{\hat{y}^\top A}_{\geq e_j^\top} \leq c^\top - e_j^\top ,$$

we see that $(\widetilde{y} - \hat{y})$ is feasible for (D) with slackness in the right-hand side in the j^{th} constraint.

Then the objective value of $\widetilde{y} - \hat{y}$ of (D) is

$$(\widetilde{y} - \hat{y})^\top b = \widetilde{y}^\top b - \hat{y}^\top b = v - \hat{y}^\top b = v$$

since $\hat{y}^\top b$ is the optimal value of D_j, which is zero.

Notice that this is just for a fixed j!

$$
\begin{array}{c|cccc|ccc}
j & & \hat{x}^\top & & & & c^\top - \hat{y}^\top A & \\
\hline
1 & \ddots & & 0 & & \ddots & & \\
\vdots & & \ddots & \vdots & & & \ddots & \\
j & \to \hat{x}^{(j)} & 0/+ & \vdots & & & +/0 & \leftarrow c^\top - \hat{y}^{(j)\top} A \\
\vdots & & & \ddots & & & \ddots & \\
n & & & 0 & \ddots & & + & \ddots \\
\hline
 & & \hat{\hat{x}} & \uparrow & & & c^\top - \hat{\hat{y}}^\top A & \\
 & & & 0 & & & + &
\end{array}
$$

Intuition. We average out for all j, then we have

$$\hat{\hat{x}} := \sum_{j=1}^{n} \frac{1}{n} \hat{x}^{(j)}, \quad \hat{\hat{y}} := \sum_{j=1}^{n} \frac{1}{n} \hat{y}^{(j)}$$

We check that $\hat{\hat{x}}$ and $\hat{\hat{y}}$ are feasible. Since

$$A\hat{\hat{x}} = A\left(\frac{1}{n}\sum_{j=1}^{n} \hat{x}^{(j)}\right) = \frac{1}{n}\sum_{j=1}^{n} \underbrace{\hat{A x}^{(j)}}_{b} = b.$$

■

Example. For multicommodity flow problem, we see that

$$
\begin{aligned}
\min \sum_{(i,j)\in\mathcal{A}} & c_{ij} x_{ij} \\
\underbrace{\sum_{j\colon (i,j)\in\mathcal{A}} x_{ij}}_{\text{flow out of } i} - & \underbrace{\sum_{j\colon (j,i)\in\mathcal{A}} x_{ji}}_{\text{flow into } i} = b_i, i \in \mathcal{N} \\
x_{ij} \ge 0 \le u_{ij} & \text{ for } (i,j) \in \mathcal{A}
\end{aligned}
$$

Proof. Write it in the matrix form, we have

$$
\begin{aligned}
\min\ & c^\top x \\
& Ax = b \\
& 0 \le \underline{x \le u},
\end{aligned}
$$

write it in another way, we have

$$
\begin{aligned}
\min\ & c^\top x \\
& Ax = b \\
& Ix \le u \\
& x \ge 0
\end{aligned}
$$

with the dual variables y and Π, we have the dual

$$\begin{aligned} \max\ & y^\top b + \Pi^\top u \\ & y^\top A + \Pi^\top I \leq c^\top \\ & y \text{ unres.},\ \Pi \leq 0. \end{aligned}$$

The A looks like

$$A_{(m\times n)} = \underset{\mathcal{N}}{\text{Nodes}} \begin{array}{cc} \begin{array}{ccccc} & & \text{arc}(i,j) & & \end{array} & \\ \begin{pmatrix} \ddots & & 0 & & \\ & \ddots & \vdots & & \\ \dots & \dots & +1 & \dots & \dots \\ & & \vdots & & \\ \dots & \dots & -1 & \dots & \dots \\ & & \vdots & \ddots & \\ & & 0 & & \ddots \end{pmatrix} & \begin{array}{c} \\ \\ i \\ \\ j \\ \\ \\ \end{array} \end{array}$$

Then we see the dual is just

$$\begin{aligned} \max\ & \sum_{i\in\mathcal{N}} y_i b_i + \sum_{(i,j)\in\mathcal{A}} \Pi_{ij} u_{ij} \\ & y_i - y_j + \Pi_{ij} \leq c_{ij} && \text{for all } (i,j)\in\mathcal{A} \\ & \Pi_{ij} \leq 0 && \text{for all } (i,j)\in\mathcal{A}. \end{aligned}$$

$\circledast$

Chapter 5

Sensitivity Analysis

Lecture 14: Sensitivity Analysis

As usual, we start with the primal and the dual 25 Oct. 08:00

$$\begin{array}{llll} & \min\ c^\top x & & \max\ y^\top b \\ & Ax = b & & y^\top A \leq c^\top. \\ (\mathrm{P}) & x \geq 0, & (\mathrm{D}) & \end{array}$$

with an optimal basic partition β, η such that

$$\overline{x} := \begin{cases} \overline{x}_\beta := A_\beta^{-1} b \geq \vec{0} \\ \overline{x}_\eta := \vec{0} \end{cases}, \qquad \overline{y}^\top := c_\beta^\top A_\beta^{-1}.$$

> **As previously seen.** The dual feasibility is
>
> $$\overline{c}_\eta := c_\eta - c_\beta^\top A_\beta^{-1} A_\eta = c_\eta - \overline{y}^\top A_\eta \geq \vec{0}$$
>
> from Lemma 3.1.2.

5.1 Local Analysis

5.1.1 Right-Hand Side Changes

We let

$$b \to b + \Delta_i e_i = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_i + \Delta_i \\ \vdots \\ b_m \end{pmatrix},$$

then

$$A_\beta^{-1}(b + \Delta_i e_i) = A_\beta^{-1} b + \Delta_i \underbrace{A_\beta^{-1} e_i}_{h^i},$$

where h_i is the i^{th} column of A_β^{-1}. So now we have

$$\overline{x}_\beta + \Delta_i h^i \geq \vec{0},$$

where we need β, η to still be an optimal partition.

5.1.2 Objective Value

Now, the objective value is

$$c_\beta^\top(\overline{x}_\beta + \Delta_i A_\beta^{-1} e_i) + c_\eta^\top \vec{0} = \underbrace{c_\beta^\top \overline{x}_\beta}_{\substack{\text{old obj.}\\ \text{value}}} + \Delta_i \underbrace{c_\beta^\top A_\beta^{-1}}_{\overline{y}^\top} e_i = c_\beta^\top \overline{x}_\beta + \Delta_i \overline{y}_i^\top.$$

5.1.3 Analysis

Let f be

$$\begin{aligned} f(b) \coloneqq \min\ & c^\top x \\ & Ax = b \\ (P_b) \quad & x \geq 0 \end{aligned}$$

where $f\colon \mathbb{R}^m \to \mathbb{R}$. We see that since the optimal objective value is equal for the dual of P_b, then $f(b) = y^\top b$. Then

$$\frac{\partial f}{\partial b_i} = \overline{y}_i$$

if $\overline{x}_\beta > \vec{0}$.

Problem. For what values of Δ_i is

$$\overline{x}_\beta + \Delta_i h^i \geq \vec{0}?$$

Answer. Firstly, we see that we need

$$\overline{x}_{\beta_K} + \Delta_i h_K^i \geq 0 \text{ for } K = 1, \ldots, m.$$

Equivalently,

$$\Delta_i h_K^i \geq -\overline{x}_{\beta_K},$$

hence

$$\begin{cases} \Delta_i \geq \dfrac{-\overline{x}_{\beta_K}}{h_K^i}, & \text{if } h_K^i > 0, \\ \Delta_i \leq \dfrac{-\overline{x}_{\beta_K}}{h_K^i}, & \text{if } h_K^i < 0. \end{cases}$$

We define L_i, U_i such that

$$L_i \leq \Delta_i \leq U_i$$

where

$$L_i \coloneqq \max_{K\colon h_K^i > 0} \{-\overline{x}_{\beta_K}/h_K^i\}, \qquad U_i \coloneqq \min_{K\colon h_K^i < 0} \{-\overline{x}_{\beta_K}/h_K^i\}.$$

⊛

Remark. Noting that if $h_K^i \leq 0$ for all K, then we define $L_i \coloneqq -\infty$. Similarly, if $h_K^i \geq 0$ for all K, we define $U_i \coloneqq \infty$.

5.2 Global Analysis

We start with a theorem.

Theorem 5.2.1. The domain of f is a convex set.

Proof. Assume that the dual of P_b is feasible, where we denote the dual as D_b:

$$\begin{aligned} \max\ & y^\top b \\ (D_b) \quad & y^\top A \leq c^\top. \end{aligned}$$

Now, the domain is the set of b such that P_b is feasible. Mathematically,

$$S := \{b \colon Ax = b, x \geq 0 \text{ are feasible.}\} \subseteq \mathbb{R}^m.$$

Suppose $b^1, b^2 \in S$. We want to check

$$\lambda b^1 + (1-\lambda)b^2 \in S \text{ for } 0 < \lambda < 1.$$

Notice that there is an x^1 such that

$$Ax^1 = b^1, x^1 \geq \vec{0}$$

and there is an x^2 such that

$$Ax^2 = b^2, x^2 \geq \vec{0}.$$

Firstly, we check that $\lambda x^1 + (1-\lambda)x^2$ is non-negative. This is clear since all components are non-negative. Then we check

$$A(\lambda x^1 + (1-\lambda)x^2) = \lambda b^1 + (1-\lambda)b^2.$$

This is clear since

$$A(\lambda x^1 + (1-\lambda)x^2) = \lambda Ax^1 + (1-\lambda)Ax^2 = \lambda b^1 + (1-\lambda)b^2.$$

∎

We now introduce the convexity of a function.

Definition 5.2.1 (Convex function). A function $f \colon S \to \mathbb{R}$ is *convex* if the domain S is convex and for all $x^1, x^2 \in S$ and $0 < \lambda < 1$,

$$f(\lambda x^1 + (1-\lambda)x^2) \leq \lambda f(x^1) + (1-\lambda)f(x^2).$$

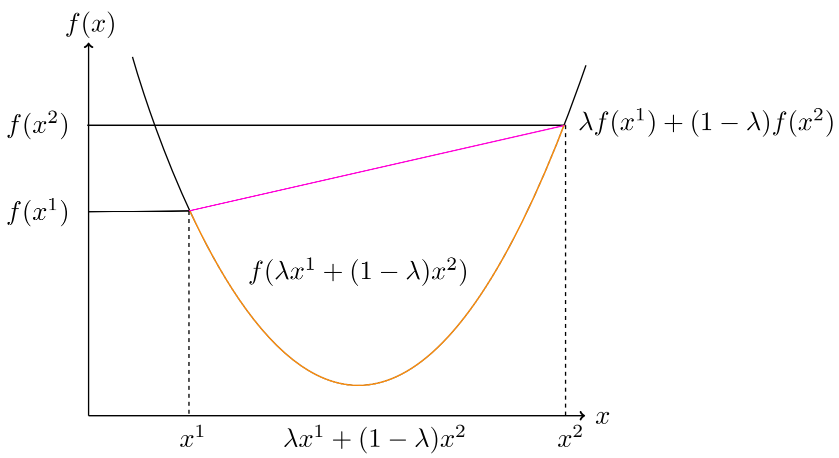


Figure 5.1: Convex function.

5.2.1 Affine Function

Before we go further, we need to have several definitions.

Definition 5.2.2 (Affine function). A function $f \colon \mathbb{R}^m \to \mathbb{R}$ is *affine* if

$$f(u_1, u_2, \ldots, u_m) = a_0 + \sum_{i=1}^{m} a_i u_i$$

where $a_i \in \mathbb{R}$ for $i = 0, \ldots, m$.

Remark. If $a_0 = 0$, then f is a linear function.

Definition 5.2.3 (Convex piece-wise linear function). A function $f\colon \mathbb{R}^m \to \mathbb{R}$ is *convex piece-wise linear* if f is the point-wise maximum of affine functions.

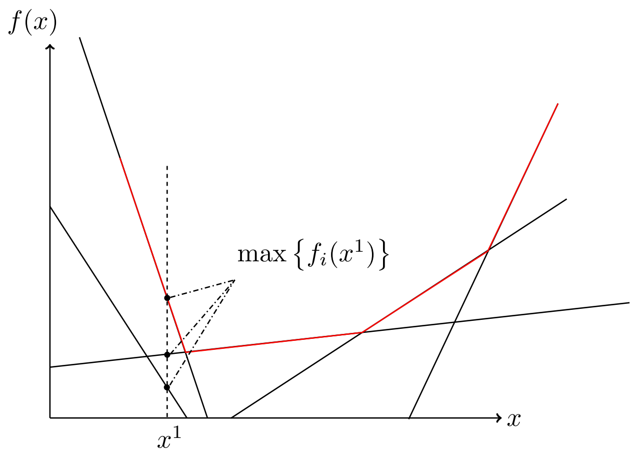


Figure 5.2: Convex piece-wise linear function.

Now, suppose $f_i\colon \mathbb{R}^m \to \mathbb{R}$ for $i = 1, \ldots, K$ and assume that each is affine. Then define

$$f(x) \coloneqq \max_{1\leq i\leq K} \{f_i(x)\}.$$

Theorem 5.2.2. The point-wise maximum of affine function is a convex function.

Proof. We see that

$$\begin{aligned}
f(\lambda x^1 + (1-\lambda)x^2) &= \max_{1\leq i\leq K} \left\{f_i(\lambda x^1 + (1-\lambda)x^2)\right\} \\
&= \max_{1\leq i\leq K} \left\{\lambda f_i(x^1) + (1-\lambda) f_i(x^2)\right\} \\
&\geq \max_{1\leq i\leq K} \left\{\lambda f_i(x^1)\right\} + \max_{1\leq i\leq K} \left\{(1-\lambda) f_i(x^2)\right\} \\
&= \lambda \max_{1\leq i\leq K} \left\{f_i(x^1)\right\} + (1-\lambda) \max_{1\leq i\leq K} \left\{f_i(x^2)\right\} = \lambda f(x^1) + (1-\lambda) f(x^2),
\end{aligned}$$

where the second equality follows from

$$\max_{1\leq i\leq K} \left\{a_{i0} + \sum_{l=1}^{m} a_{il}(\lambda u_l^1 + (1-\lambda)u_l^2)\right\} = \max_{1\leq i\leq K} \left\{\lambda a_{i0} + (1-\lambda)a_{i0} + \sum_{l=1}^{m} a_{il}(\lambda u_l^1 + (1-\lambda)u_l^2)\right\}.$$

■

Lecture 15: Sensitivity Analysis

5.3 More on Local Analysis

27 Oct. 08:00

As previously seen. Based on an optimal basic solution:

$$\overline{x}_\beta \coloneqq A_\beta^{-1}\boldsymbol{b} \geq \vec{0}$$

and the reduced cost

$$\bar{c}_\eta := \boldsymbol{c}_{\boldsymbol{\eta}}^\top - \boldsymbol{c}_{\boldsymbol{\beta}}^\top A_\beta^{-1} \boldsymbol{A}_{\boldsymbol{\eta}} \geq \vec{0},$$

we see that $\boldsymbol{c}, \boldsymbol{b}, \boldsymbol{A}_{\boldsymbol{\eta}}$ are linear respect to the objective value. Therefore, there is no limitation for us to only do local analysis respect to b, we can do this for any one of the data mentioned above.

5.3.1 Data Changes

We now change A_η to do the local analysis for example. If

$$a_{i,\eta_j} \to a_{i,\eta_j} + \Delta,$$

then

$$A_{\eta_j} = \begin{pmatrix} a_{1,\eta_j} \\ a_{2,\eta_j} \\ \vdots \\ a_{m,\eta_j} \end{pmatrix}.$$

Problem. For what Δ is β, η still an optimal partition?

Answer. We see that the reduced cost is now

$$\bar{c}'_{\eta_j} = c_{\eta_j} - \underline{c_\beta^\top A_\beta^{-1}} \left(A_{\eta_j} + \Delta e_i\right) = c_{\eta_j} - \overline{y}^\top (A_{\eta_j} + \Delta e_i) = \bar{c}_{\eta_j} - \Delta \overline{y}_i \underset{\text{want}}{\geq} 0.$$

Hence, the condition becomes

$$\bar{c}_{\eta_j} \geq \Delta \overline{y}_i.$$

⊛

5.3.2 Objective Coefficients Changes

We can also try to change c for local analysis. Firstly, consider changing c_{η_j}, we have

$$c_{\eta_j} \to c_{\eta_j} + \Delta,$$

then the reduced cost for x_{η_j} becomes

$$(c_{\eta_j} + \Delta) - \overline{y}^\top A_{\eta_j} = \bar{c}_{\eta_j} + \Delta \underset{\text{want}}{\geq} 0.$$

Hence, the condition becomes

$$\Delta \geq -\bar{c}_{\eta_j}.$$

Now, for c_{β_i},

$$c_{\beta_i} \to c_{\beta_i} + \Delta.$$

Then

$$\underline{c_\eta^\top} - (\underline{c_\beta^\top} + \Delta e_i^\top)\underline{A_\beta^{-1} A_\eta} \underset{\text{want}}{\geq} \vec{0}.$$

We see that the underlined part is just $\bar{c}_\eta$, hence the reduced cost is just

$$\bar{c}_\eta^\top - \Delta e_i^\top \overline{A}_\eta = (\bar{c}_{\eta_1}, \dots, \bar{c}_{\eta_{n-m}}) - \Delta(\bar{a}_{i,\eta_1}, \bar{a}_{i,\eta_2}, \dots, \bar{a}_{i,\eta_{n-m}}) \underset{\text{want}}{\geq} 0$$

Separate them, we see

$$\bar{c}_{\eta_j} - \Delta \bar{a}_{i,\eta_j} \geq 0 \text{ for } j = 1, \dots, n-m.$$

Equivalently,

$$\Delta \leq \frac{\bar{c}_{\eta_j}}{\bar{a}_{i,\eta_j}} \text{ for } j \text{ such that } \bar{a}_{i,\eta_j} > 0$$

and

$$\Delta \geq \frac{\bar{c}_{\eta_j}}{\bar{a}_{i,\eta_j}} \text{ for } j \text{ such that } \bar{a}_{i,\eta_j} < 0.$$

Recall the definition of L and U, we can have the similar inequality for Δ such that $L \leq \Delta \leq U$, where

$$L := \max_{j\colon \bar{a}_{i,\eta_j} < 0} \left\{ \frac{\bar{c}_{\eta_j}}{\bar{a}_{i,\eta_j}} \right\} \leq \Delta \leq \min_{j\colon \bar{a}_{i,\eta_j} > 0} \left\{ \frac{\bar{c}_{\eta_j}}{\bar{a}_{i,\eta_j}} \right\} =: U.$$

5.3.3 Right-Hand Side Changes – Two Entries

There is no limitation for us to change two entries for b. Consider

$$b \to b + \Delta(e_i - e_K).$$

Then

$$\overline{x}'_\beta = A_\beta^{-1}(b + \Delta(e_i - e_K)) = \overline{x}_\beta + \Delta A_\beta^{-1}(e_i - e_K) = \overline{x}_\beta + \Delta(h_i - h_K) \underset{\text{want}}{\geq} \vec{0}.$$

Writing things separately, we have

$$\overline{x}_{\beta_l} + \Delta(h_{il} - h_{Kl}) \geq 0 \text{ for } l = 1 \ldots, m$$

where $H \coloneqq A_\beta^{-1}$. Then,

$$\Delta \geq \frac{-\overline{x}_{\beta_l}}{h_{il} - h_{Kl}} \text{ if } h_{il} - h_{Kl} > 0$$

and

$$\Delta \leq \frac{-\overline{x}_{\beta_l}}{h_{il} - h_{Kl}} \text{ if } h_{il} - h_{Kl} < 0.$$

But when we want to change more than one variable in the same time, it becomes more complicated. Consider

$$b \to b + \Delta_i e_i, \qquad c_{\beta_l} \to c_{\beta_l} + \Delta_l e_l.$$

The condition for β, η still being a basic partition is

$$A_\beta^{-1}(b + \Delta_i e_i) \geq \vec{0}, \qquad c_\eta - (c_\beta + \Delta_l e_l)^\top A_\eta \geq \vec{0}.$$

Originally, the objective value is

$$c_\beta^\top (A_\beta^{-1} b) = c_\beta^\top \overline{x}_\beta = (c_\beta^\top A_\beta^{-1}) b = \overline{y}^\top b,$$

after considering the changes, we have

$$(c_\beta + \Delta_l e_l)^\top A_\beta^{-1}(b + \Delta_i e_i).$$

We see that this is a *quadratic* relation. Expanding the expression, we have

$$c_\beta^\top A_\beta^{-1} b + \Delta_i c_\beta^\top A_\beta^{-1} e_i + \Delta_l e_l^\top A_\beta^{-1} b + \Delta_i \Delta_l e_l^\top A_\beta^{-1} e_i = c_\beta A_\beta^{-1} b + \Delta_i \overline{y}_i + \Delta_l \overline{x}_{\beta_l} + \Delta_i \Delta_l h_{li}$$

where again, $H \coloneqq A_\beta^{-1}$.

> **Remark.** We see that if we hold one of Δ_i or Δ_l being 0, it's still a linear relation.

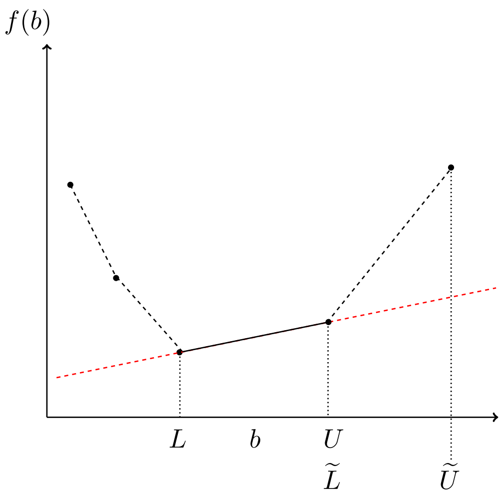


Figure 5.3: Local Analysis

5.4 More on Global Analysis

Still, consider the primal and dual pair

$$
\begin{array}{rlrl}
f(b) = \min & c^\top x & \max & y^\top b \\
 & Ax = b & & y^\top A \leq c^\top. \\
(P_b) & x \geq 0 & (D_b) &
\end{array}
$$

A basis β is feasible for D_b is independent of b. (recall that $\overline{y}^\top := c_\beta^\top A_\beta^{-1}$) Then we have

$$f(b) := \max \left\{ (c^\top A^{-1})_\beta b \colon \beta \text{ is a dual feasible basis} \right\}.$$

Consider

$$
\begin{array}{rl}
g(c) = \min & c^\top x \\
 & Ax = b \\
(P_c) & x \geq 0
\end{array}
$$

where g is a piece-wise linear concave function (contrast to Definition 5.2.3) in c. We see that D_b is equivalence to

$$
\begin{array}{rl}
-\min & -(y^+ - y^-)^\top b \\
 & (y^+ - y^-)^\top A + IS^\top = c^\top \\
 & y^+ \geq 0,\ y^- \geq 0.
\end{array}
$$

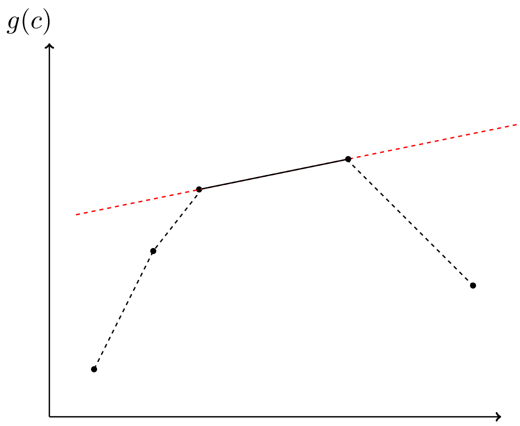


Figure 5.4: The dual version

Chapter 6

Large-Scale Linear Optimization

Lecture 16: Large-Scale Linear Optimization

Let's first look at an example. 1 Nov. 08:00

Example (Nearly separated matrix). Given a *nearly separated* constraint matrix, i.e.,

$$A = \begin{pmatrix} [\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad] \\ \begin{bmatrix} \quad \\ \quad \end{bmatrix} & & & & 0 & \\ & [\quad] & & & & \\ & & \begin{bmatrix} \quad \\ \quad \\ \quad \end{bmatrix} & & & \\ & 0 & & & \ddots & \\ & & & & & \begin{bmatrix} \quad \\ \quad \end{bmatrix} \end{pmatrix},$$

then if the first constraint (the first row) doesn't exist, the corresponding linear program is easy.

Proof. In this case, then the problem decomposes to those small block matrix corresponds to some smaller, easier linear optimization problems, and we can solve it very quickly. ⊛

There is something we need in order to solve the above problem.

6.1 Decomposition Algorithm

In this section we describe what is usually known as **Dantzig-Wolfe Decomposition**. We need

1. Simplex Algorithm.
2. Geometry of basic feasible solutions and directions.
3. Duality.

We first see a useful theorem.

6.1.1 Representation Theorem

Let (P) be

$$\begin{aligned} \min\ & c^\top x \\ & Ax = b \\ \text{(P)}\quad & x \geq 0. \end{aligned}$$

Theorem 6.1.1 (Representation theorem)**.** Suppose that (P) is feasible. Then let $\mathcal{X}$ be

$$\mathcal{X} := \{\hat{x}^j : j \in \mathcal{J}\}$$

be the set of basic feasible solutions of (P). Also, let $\mathcal{Z}$ be

$$\mathcal{Z} := \{\hat{z}^k : k \in \mathcal{K}\}$$

be the set of basic feasible rays of (P). Then the feasible region of (P) is equal to

$$S' := \left\{ \sum_{j\in\mathcal{J}} \lambda_j \hat{x}^j + \sum_{k\in\mathcal{K}} \mu_k \hat{z}^k : \sum_{j\in\mathcal{J}} \lambda_j = 1;\ \lambda_j \geq 0,\ j \in \mathcal{J};\ \mu^k \geq 0,\ k \in \mathcal{K} \right\}.$$

Proof. Let S be the feasible region of (P). We show that $S = S'$ by showing $S' \subseteq S$ and $S' \supseteq S$.

1. $S' \subseteq S$. Since

$$A\left(\sum_j \lambda_j \hat{x}^j + \sum_K \mu_K \hat{z}^K\right) = \sum_j \lambda_j \underbrace{\left(A\hat{x}^j\right)}_{=b} + \sum_K \mu^K \underbrace{\left(A\hat{z}^K\right)}_{=0} = b.$$

   Moreover, since everything in the sum is non-negative, we see that $S' \subseteq S$.

2. $S \subseteq S'$. Assume $\hat{x} \in S$. Then consider the following system

$$\underset{\text{equations}}{\overset{n+1}{}}\begin{cases} \displaystyle\sum_{j\in\mathcal{J}} \lambda_j \hat{x}^j + \sum_{k\in\mathcal{K}} \mu_k \hat{z}^k & = \hat{x} \\ \displaystyle\sum_j \lambda_j & = 1 \end{cases}$$

$$\text{(I)} \quad \lambda_j \geq 0 \text{ for } j \in \mathcal{J};\ \mu^k \geq 0 \text{ for } k \in \mathcal{K}.$$

> **Note.** Keep in mind that in the above system, $\hat{x}$ and $\hat{z}$ are fixed, the variables are the λ_j and μ_k.

   Now, instead of directly constructing a solution, we use Farkas Lemma. Namely, we write down a system such that if this system is infeasible, by Farkas Lemma, our original system is feasible. Firstly, in Farkas Lemma, we have

$$A = \begin{pmatrix} \hat{x}^1 & \hat{x}^2 & \dots & \hat{z}^1 & \hat{z}^2 & \hat{z}^3 & \dots \\ 1 & 1 & \dots & 0 & 0 & \dots & 0 \end{pmatrix}, \qquad b = \begin{pmatrix} \hat{x} \\ 1 \end{pmatrix}$$

   in (I). Now, denote the dual variables with w, t, then we have

$$\begin{pmatrix} w^\top & t \end{pmatrix} \begin{pmatrix} \hat{x} \\ 1 \end{pmatrix} > 0$$

$$\begin{pmatrix} w^\top & t \end{pmatrix} \begin{pmatrix} \hat{x}^j \\ 1 \end{pmatrix} \leq 0 \text{ for } j \in \mathcal{J}$$

$$\begin{pmatrix} w^\top & t \end{pmatrix} \begin{pmatrix} \hat{z}^k \\ 0 \end{pmatrix} \leq 0 \text{ for } k \in \mathcal{K}$$

   for (II). We only need to show that (II) cannot have a solution. This is easy to show. Firstly, we see that the above inequalities are equivalent to

$$\begin{aligned} w^\top \hat{x} + t &> 0 & -w^\top \hat{x} &< t \\ w^\top \hat{x}^j + t &\leq 0 \Leftrightarrow & -w^\top \hat{x}^j &\geq \hat{t} \text{ for } j \in \mathcal{J} \\ w^\top \hat{z}^k &\leq \vec{0} & -w^\top \hat{z}^K &\geq 0 \text{ for } k \in \mathcal{K}. \end{aligned}$$

Now, suppose this does have a solution $\hat{w}, \hat{t}$. Then, consider

$$\begin{aligned} \min \quad & -\hat{w}^\top x (< \hat{t}) \\ & Ax = b \\ & x \geq 0. \end{aligned}$$

Notice that the objective value of $\hat{x}$ here is less than $\hat{t}$ by (II). Since we know that $Ax = b$, hence this linear programming is feasible. Moreover, from $-\hat{w}^\top \hat{x} \leq \hat{t}$ and $-\hat{w}^\top \hat{x}^j \geq \hat{t}$, we see that we have a better solution with respect to the objective function among the linear combination of extreme points $\hat{x}^j$. But this is only possible for unbounded linear programming problem, which needs the positive dot product between rays and the objective vector. But from $-\hat{w}\hat{z}^k \geq 0$, we see that this will never happen, hence the theorem is proved.

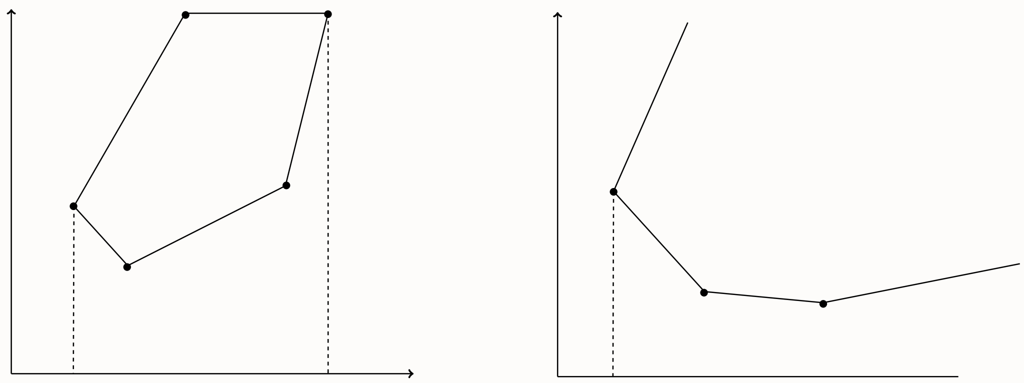


Figure 6.1: Bounded and unbounded case in Simplex Algorithm

■

With this representation theorem, consider

$$\begin{aligned} \min \quad & c^\top x \\ & Ex \geq h \\ \text{"easy"} & \begin{cases} Ax = b \\ x \geq 0. \end{cases} \end{aligned}$$

Then by

$$\Bigg\{ \underbrace{\sum_{j \in \mathcal{J}} \lambda_j \hat{x}^j + \sum_{k \in \mathcal{K}} \mu_k \hat{z}^k}_{= \{x \in \mathbb{R}^n :\, Ax = b,\ x \geq \vec{0}\}} : \sum_{j \in \mathcal{J}} \lambda_j = 1, \lambda_j \geq 0 \text{ for } j \in \mathcal{J}, \mu^k \geq 0 \text{ for } k \in \mathcal{K} \Bigg\},$$

we turn the linear problem into

$$
\begin{aligned}
\min\ & c^\top \left(\sum_{j\in\mathcal{J}} \lambda_j \hat{x}^j + \sum_{k\in\mathcal{K}} \mu_k \hat{z}^k \right) \\
& E\left(\sum_{j\in\mathcal{J}} \lambda_j \hat{x}^j + \sum_{k\in\mathcal{K}} \mu_k \hat{z}^k \right) \geq h \\
& \sum_{j\in\mathcal{J}} \lambda_j = 1 \\
& \lambda_j \geq 0 \text{ for } j\in\mathcal{J},\ \mu_K \geq 0 \text{ for } k\in\mathcal{K}.
\end{aligned}
$$

Furthermore, this is equivalent to

$$
\begin{aligned}
\min\ & \sum_{j\in\mathcal{J}} \left(c^\top \hat{x}^j\right) \lambda_j + \sum_{k\in\mathcal{K}} \left(c^\top \hat{z}^k\right) \mu_k \\
& \sum_{j\in\mathcal{J}} \left(E\hat{x}^j\right) \lambda_j + \sum_{k\in\mathcal{K}} \left(E\hat{z}^k\right) \mu_k \geq h \\
& \sum_{j\in\mathcal{J}} \lambda_j = 1 \\
(\mathrm{M})\quad & \lambda_j \geq 0 \text{ for } j\in\mathcal{J},\ \mu_k \geq 0 \text{ for } k\in\mathcal{K}.
\end{aligned}
$$

The system is now extremely reduced, but the cost is that we now have huge amount of variables. We call this as the main problem.

Definition 6.1.1 (Main problem). Given a linear programming problem

$$
\begin{aligned}
\min\ & c^\top x \\
& Ex \geq h \\
\text{"easy"} & \begin{cases} Ax = b \\ x \geq 0, \end{cases}
\end{aligned}
$$

the so-called *main problem* is defined as

$$
\begin{aligned}
\min\ & \sum_{j\in\mathcal{J}} \left(c^\top \hat{x}^j\right) \lambda_j + \sum_{k\in\mathcal{K}} \left(c^\top \hat{z}^k\right) \mu_k \\
& \sum_{j\in\mathcal{J}} \left(E\hat{x}^j\right) \lambda_j + \sum_{k\in\mathcal{K}} \left(E\hat{z}^k\right) \mu_k \geq h \\
& \sum_{j\in\mathcal{J}} \lambda_j = 1 \\
(\mathrm{M})\quad & \lambda_j \geq 0 \text{ for } j\in\mathcal{J},\ \mu_k \geq 0 \text{ for } k\in\mathcal{K}.
\end{aligned}
$$

We formalize the above result as so-called Decomposition Theorem.

Theorem 6.1.2 (Decomposition theorem). Let

$$
\begin{aligned}
\min\ & c^\top x \\
& Ex \geq h \\
& Ax = b \\
(Q)\quad & x \geq 0
\end{aligned}
$$

Let $S := \{x\in\mathbb{R}^n : Ax = b, x\geq 0\}$, $\mathcal{X} := \left\{\hat{x}^j : j\in\mathcal{J}\right\}$ be the set of basic feasible solutions S and $\mathcal{Z} := \left\{\hat{z}^k : k\in\mathcal{K}\right\}$ be the set of basic feasible rays of S. Then Q is equivalent to the main problem

(M)

$$
\begin{aligned}
\min \quad & \sum_{j\in\mathcal{J}} \left(c^\top \hat{x}^j\right)\lambda_j + \sum_{k\in\mathcal{K}} \left(c^\top \hat{z}^k\right)\mu_k \\
& \sum_{j\in\mathcal{J}} \left(E\hat{x}^j\right)\lambda_j + \sum_{k\in\mathcal{K}} \left(E\hat{z}^k\right)\mu_k \geq h \\
& \sum_{j\in\mathcal{J}} \lambda_j = 1 \\
\text{(M)} \quad & \lambda_j \geq 0 \text{ for } j\in\mathcal{J},\ \mu_k \geq 0 \text{ for } k\in\mathcal{K}.
\end{aligned}
$$

Remark. We think of E being a *complicated* constraint matrix, while A is much easier. Further, the reason why we choose $\leq$ for E and $=$ for A is not because this makes them complicated or easy, but only for our convenience. In deed, we will soon see that we can turn M into a standard form problem without increasing complexity.

6.2 Solution of the Master Problem via the Simplex Algorithm

We now want to solve (M). And since we can't write out (M) explicitly since there are too many variables. But instead, we can reasonably *maintain* a basic solution of ($\overline{\text{M}}$), the standard form of (M). Furthermore, the only part of the simplex algorithm that is sensitive to the total number of variables is when we check for variables with negative reduced cost. So we now try to find an indirect way to check this rather than find it one by one.

Denotes the dual variable of (M) as y and σ with $y \geq \vec{0}$ and σ unrestricted. We further turn (M) into the standard form problem, which is just

$$
\begin{aligned}
\min \quad & \sum_{j\in\mathcal{J}} \left(c^\top \hat{x}^j\right)\lambda_j + \sum_{k\in\mathcal{K}} \left(c^\top \hat{z}^k\right)\mu_k \\
& \sum_{j\in\mathcal{J}} \left(E\hat{x}^j\right)\lambda_j + \sum_{k\in\mathcal{K}} \left(E\hat{z}^k\right)\mu_k - Is = h \\
& \sum_{j\in\mathcal{J}} \lambda_j = 1 \\
(\overline{\text{M}}) \quad & \lambda_j \geq 0 \text{ for } j\in\mathcal{J},\ \mu_K \geq 0 \text{ for } k\in\mathcal{K}, s \geq 0.
\end{aligned}
$$

Suppose that $\overline{y}$, $\overline{\sigma}$ forms a basic dual solution. The reduced cost of λ_j associated with $\hat{x}^j$ is

$$
(c^\top \hat{x}^j) - \begin{pmatrix}\overline{y}^\top & \overline{\sigma}\end{pmatrix} \begin{pmatrix} E\hat{x}^j \\ 1 \end{pmatrix} = c^\top \hat{x}^j - \hat{y}^\top E\hat{x}^j - \overline{\sigma} = (c^\top - \overline{y}^\top E)\hat{x}^j - \overline{\sigma}
$$

since $\overline{c}_{\eta_j} = c_{\eta_j} - \overline{y}^\top A_{\eta_j}$.

Problem 6.2.1. Is there a λ_j with this reduced cost negative?

Answer. Consider

$$
\begin{aligned}
-\sigma + \min \quad & (c^\top - \overline{y}^\top E)x \\
& Ax = b \\
& x \geq 0.
\end{aligned}
$$

⊛

Lecture 17: Large-Scale Linear Optimization

As previously seen. We now focus on one particular problems: What's the conditions for a variable to enter the basis? 3 Nov. 08:00

(a) What's the reduced coast of s_i?

$$0 - \begin{pmatrix}\overline{y}^\top & \overline{\sigma}\end{pmatrix}\begin{pmatrix}-e_i\\0\end{pmatrix} = \overline{y}_i.$$

If $\overline{y}_i < 0$, then s_i can enter the basis.

(b) What's the reduced cost of λ_j?

$$(c^\top \hat{x}^j) - \begin{pmatrix}\overline{y}^\top & \overline{\sigma}\end{pmatrix}\begin{pmatrix}E\hat{x}^j\\1\end{pmatrix} = c^\top \hat{x}^j - \hat{y}^\top E\hat{x}^j - \overline{\sigma} = (c^\top - \overline{y}^\top E)\hat{x}^j - \overline{\sigma}.$$

We consider a sub problem

$$\begin{aligned} -\sigma + \min\ & (c^\top - \overline{y}^\top E)x \\ & Ax = b \\ \text{(SUB)}\quad & x \geq 0. \end{aligned}$$

If the optimal values < 0, then the optimal basic solution $\hat{x}^j$ has an associated λ_j with negative reduced cost, so λ_j can enter the basis of (M). Else if the optimal value ≥ 0, then no λ_j can enter the basis.

Note. We need to include $-\sigma$ for evaluating the optimal values.

Problem. What if the optimal value is unbounded?

(c) What's the reduced cost of μ^k?

$$(c^\top \hat{z}^k) - \begin{pmatrix}\overline{y}^\top & \overline{\sigma}\end{pmatrix}\begin{pmatrix}E\hat{z}^k\\0\end{pmatrix} = (c^\top - \overline{y}^\top E)\hat{z}^k.$$

Again, consider a sub problem

$$\begin{aligned} \min\ & (c^\top - \overline{y}^\top E)z \\ & Az = \vec{0} \\ & z \geq 0. \end{aligned}$$

Remark. Compare this problem to the previous sub problem (SUB).

(a) Notice that the objective value of this problem will always be 0 or unbounded. Since 0 is always a feasible solution, or if once it's negative, we can multiply it by a positive number and make the optimal values smaller.

(b) When solving (SUB), the optimal values of (SUB) is

i. negative $\Rightarrow \lambda_j$ to enter the basis;

ii. non-negative $\Rightarrow$ no λ_j can enter the basis;

iii. unbounded $\Rightarrow$ we get a $\overline{z}$ that is a basic ray with $c^\top \overline{z} < 0$, which implies for some $\hat{z}^k$, μ^k with negative reduced cost.

Note. We stop when (SUB) has the optimal values being 0.

Why the optimal values of (SUB) will always be non-positive?

Now, we know what variable can enter the basis, but we have not yet consider what variable can leave. Recall that the basic matrix B for $(\overline{\text{M}})$ will have the following columns

$$s_i = \begin{pmatrix}0\\1\\\vdots\\0\\\mathbf{0}\end{pmatrix}, \qquad \lambda_j = \begin{pmatrix}E\hat{x}^j\\ \\ \mathbf{1}\end{pmatrix}, \qquad \mu^k = \begin{pmatrix}E\hat{z}^k\\ \\ \mathbf{0}\end{pmatrix},$$

where we see that the last entries of λ_j will always be 1, and at least one of λ_j will be in the basis due to the fact that B is invertible. For simplicity, we just consider

$$B = \begin{array}{c} \begin{pmatrix} & -I & & E\hat{x}^1 \\ 0 & \dots & 0 & 1 \end{pmatrix} \\ \scriptstyle s_1 \; s_2 \;\; \dots \; s_k \quad \lambda_1 \end{array}$$

where we get $\hat{x}^1$ by solving

$$\begin{aligned} \min \ & e^\top \hat{x} \\ & Ax = b \\ & x \geq 0 \end{aligned}$$

If $E\hat{x}^1 \geq h$, then $\overline{s} \geq \vec{0} \Rightarrow$ directly go to Phase II. Then,

$$\begin{pmatrix} \overline{y}^\top & \overline{\sigma} \end{pmatrix} = \begin{pmatrix} (\overline{c}\hat{x}^j) & (c^\top \hat{z}^k) & 0 \end{pmatrix} B^{-1},$$

where $\overline{c}\hat{x}^j$ initially is

$$\begin{pmatrix} 0 & \dots & 0 & c^\top \hat{x}^1 \end{pmatrix}.$$

Recall the ratio test for determining what entry should enter the basis and what should leave. Namely,

$$\overline{y}^\top = c_\beta^\top A_\beta^{-1}, \quad \overline{x}_\beta = A_\beta^{-1} b = \begin{pmatrix} \overline{x}_{\beta_1} \\ \vdots \\ \overline{x}_{\beta_m} \end{pmatrix}, \quad \overline{A}_{\eta_j} = A_\beta^{-1} A_{\eta_j} = \begin{pmatrix} \overline{a}_{1,\eta_j} \\ \vdots \\ \overline{a}_{m,\eta_j} \end{pmatrix}$$

with the ratio being

$$\min_{i \colon \overline{a}_{i\eta_j} > 0} \left\{ \frac{\overline{x}_{\beta_i}}{\overline{a}_{i,\eta_j}} \right\}.$$

Now, in our situation, we carry out the ratio test by noting that the basic variable values is just

$$B^{-1} \begin{pmatrix} h \\ 1 \end{pmatrix},$$

and the updated entering column is

$$B^{-1} \begin{pmatrix} -e_i \\ 0 \end{pmatrix} \text{ or } B^{-1} \begin{pmatrix} E\hat{x}^j \\ 1 \end{pmatrix} \text{ or } B^{-1} \begin{pmatrix} E\hat{z}^k \\ 0 \end{pmatrix},$$

which corresponds to λ_j, μ_k, s_i is entering the basis, respectively.

Then we just do the ratio test. If $B^{-1} \begin{pmatrix} h \\ 1 \end{pmatrix} \geq \vec{0} \Rightarrow$ go to Phase II. If not we create an artificial column

$$\begin{pmatrix} E\hat{x}^1 \\ 1 \end{pmatrix}.$$

Lecture 18: Lagrangian Relaxation

8 Nov. 08:00

As previously seen. The Simplex Algorithm.

1. Initialization (Phase I). Find an initial basic feasible partition β, η
2. Is there a non-basis variable with negative reduced cost?

$$\overline{c}_j \coloneqq c_j - \overline{y}^\top A_{\eta_j} < 0.$$

   If not, then we have an optimal solution.

3. Find the leaving variable.

$$i^* \coloneqq \arg \max_{\overline{a}_{i,\eta_j} > 0} \left\{ \frac{\overline{x}_{\beta_i}}{\overline{a}_{i,\eta_j}} \right\}.$$

   If i^* is undefined, then problem is unbounded.

4. Swap β_i and η_j and **GOTO 2**.

Then the decomposition algorithm can be written as follows. We change the step 0. and 2. of the above simplex algorithm into the following.

0. Reformulate Q as M and apply simplex algorithm to M, where

$$\begin{aligned} \min \ & c^\top x \\ & Ex \geq h \\ & Ax = b \\ (Q) \quad & x \geq 0 \end{aligned}$$

and

$$\begin{aligned} \min \ & \sum_{j\in\mathcal{J}} \left(c^\top \hat{x}^j\right) \lambda_j + \sum_{k\in\mathcal{K}} \left(c^\top \hat{z}^k\right) \mu_k \\ & \sum_{j\in\mathcal{J}} \left(E\hat{x}^j\right) \lambda_j + \sum_{k\in\mathcal{K}} \left(E\hat{z}^k\right) \mu_k \geq h \\ & \sum_{j\in\mathcal{J}} \lambda_j = 1 \\ (\mathrm{M}) \quad & \lambda_j \geq 0 \text{ for } j \in \mathcal{J}, \ \mu_k \geq 0 \text{ for } k \in \mathcal{K}. \end{aligned}$$

2. Solve the sub-problem

$$\begin{aligned} -\overline{\sigma} + \min \ & \overline{c} - \overline{y}^\top E \\ & Ax = b \\ & x \geq 0 \end{aligned}$$

- optimal & Objective value $< 0 \Rightarrow$ a λ variable can enter the basis.
- optimal & Objective value $> 0 \Rightarrow$ have an optimal for (M).
- Unbounded $\Rightarrow$ a μ_k variable can enter the basis.

Note. Compare 2. here and 2. in the simplex algorithm.

Remark. For the real implementation in step 2., we

1. Keep all generated columns.
2. First check reduced costs of columns already generated. **Repeat**. Only solve for sub-problem when needed.

Note. We see that we are solving (M) over the known columns. So instead, we can pass (M) to a solver (`Gurobi`). And since it will give us the dual variable $\overline{y}$ and $\overline{\sigma}$, we can continue to solve the sub-problem without problems. Furthermore, we solve the sub-problem and append new column to known ones and go solve the sub-problem again. In short, let the solver keep track of the basis.

6.3 Lagrangian Relaxation

The motivation is to get a good lower bound of optimal objective value for

$$\begin{aligned} z := \min \ & c^\top x \\ & Ex \geq h \\ & Ax = b \\ (Q) \quad & x \geq 0 \end{aligned}$$

Since the problem is large, hence we want to exit the algorithm whenever we get a *good enough* solution such that it's not far away from the objective value. But the problem is, when should we stop? Do we stop at plateaus? What if there is a second drop in terms of objective value?

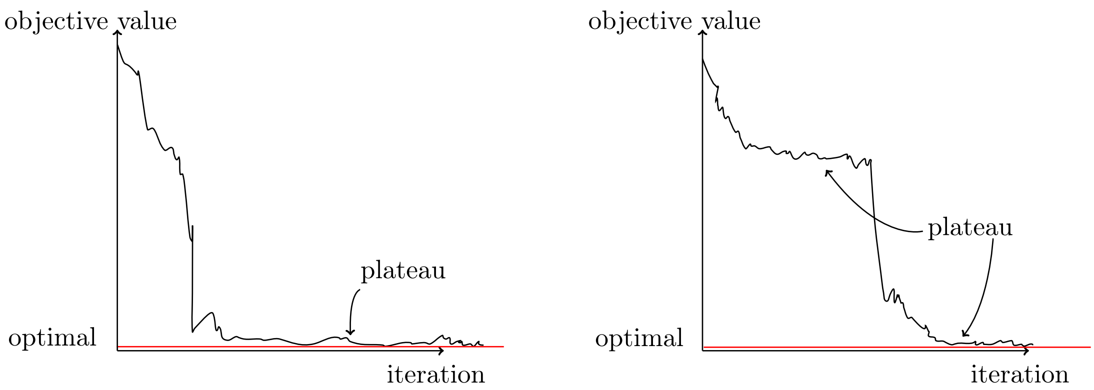


Figure 6.2: Early Arrival, can we?

6.3.1 Lagrangian Bounds

We first start with a specific problem which is important in our analysis.

Definition 6.3.1 (Lagrangian subproblem)**.** We choose $\hat{y} \geq \vec{0}$, and the corresponding *Lagrangian subproblem* $L_{\hat{y}}$ is defined as

$$\begin{aligned} v(\hat{y}) := \hat{y}^\top h + \min\ & (c^\top - \hat{y}^\top E)x \\ & Ax = b \\ (L_{\hat{y}}) \quad & x \geq 0 \end{aligned}$$

where L stands for Lagrange.

Intuition. We are trying to *bring* the complex constraint $Ex \geq h$ into the objective function.

To characterize how good will this approximation be, we first see a simple result.

Lemma 6.3.1. For any $\hat{y} \geq \vec{0}$, $v(\hat{y}) \leq z$.

Proof. Let x^* be an optimal solution for Q. Then x^* is feasible for $L_{\hat{y}}$. Then we see

$$v(\hat{y}) \leq \hat{y}^\top h + (c^\top - \hat{y}^\top E)x^* = \underbrace{c^\top x^*}_{z} + \underbrace{\hat{y}^\top}_{\geq \vec{0}} \underbrace{(h - Ex^*)}_{\leq \vec{0}} \leq z$$

■

Denote the dual variables of Q as y and π. The dual of Q is

$$\begin{aligned} \max\ & y^\top h + \pi^\top b \\ & y^\top E + \pi^\top A \leq c^\top \\ & y \geq 0 \end{aligned}$$

and the dual of $L_{\hat{y}}$ is

$$\begin{aligned} \hat{y}^\top h + \max\ & \pi^\top b \\ & \pi^\top A \leq c^\top - \hat{y}^\top E \end{aligned}$$

Note. $\hat{y}$ is not the variable.

Theorem 6.3.1 (Laguargian dual theorem). Suppose x^* is optimal for Q. Further, suppose $\hat{y}$ and $\hat{\pi}$ are optimal for the dual of Q. Then

- x^* is optimal for $L_{\hat{y}}$
- $\hat{\pi}$ is optimal for the dual of $L_{\hat{y}}$
- $\hat{y}$ is a maximizer of $v(y)$ over $y > \vec{0}$
- The maximum value of $v(y)$ over $y \geq \vec{0}$ is z.

Proof. We first make a note.

Note. In above, we want to find
$$z = \max_{y \geq 0} \ v(y).$$

x^* is feasible for $L_{\hat{y}}$ and $\hat{y}$ and $\hat{\pi}$ is feasible for the dual of Q. Then

$$\hat{y}^\top E + \hat{\pi}^\top A \leq c^\top$$

with $\hat{y}^\top \geq \vec{0}$. But we see that this is equivalent to

$$\hat{\pi}^\top A \leq c^\top - \hat{y}^\top E,$$

which implies $\hat{\pi}$ is feasible for the dual of $L_{\hat{y}}$.

From strong duality theorem for Q,

$$c^\top x^* = \hat{y}^\top h + \hat{\pi}^\top b.$$

Then, by using $E\hat{x}^* \geq h$, we see that

$$(c^\top - \hat{y}^\top E)x^* \leq \hat{\pi}^\top b.$$

Moreover, recall $\hat{\pi}$ is feasible for the dual of $L_{\hat{y}}$ with $\hat{\pi}^\top A \leq c^\top - \hat{y}^\top E$, then since $x^* > 0$, we have

$$\hat{\pi}^\top \underbrace{Ax^*}_{b} \leq (c^\top - \hat{y}^\top E)x^* \Leftrightarrow \hat{\pi}^\top b \leq (c^\top - \hat{y}^\top E)x^*.$$

We conclude
$$\hat{\pi}^\top b = (c^\top - \hat{y}^\top E)x^*.$$

Now, we claim that x^* is optimal for $L_{\hat{y}}$ and $\hat{\pi}$ is optimal for the dual of $L_{\hat{y}}$. Indeed, recall that the objective function of $L_{\hat{y}}$ is
$$\hat{y}^\top h + \min \ (c^\top - \hat{y}^\top E)x,$$

with $(c^\top - \hat{y}^\top E)x^* \geq \hat{\pi}^\top b$, we see that the objective value of x^* in $L_{\hat{y}}$ is equal to $\hat{y}h + \hat{\pi}^\top b$, which implies that x^* is optimal in $L_{\hat{y}}$ from weak duality theorem.

Lastly, since x^* is optimal for $L_{\hat{y}}$,

$$z \geq v(\hat{y}) = \hat{y}^\top h + (c^\top - \hat{y}^\top E)x^* = \hat{y}^\top h + \hat{\pi}^\top b = z$$

by optimally for $\hat{y}$ and $\hat{\pi}$, hence we see that $\hat{y}$ solves

$$\max_{y \geq 0} \ v(y)$$

and $v(\hat{y}) = z$. ∎

Conversely, we have the following.

Theorem 6.3.2 (Converse Lagrangian dual theorem). Suppose that $\hat{y}$ is a maximizer of $v(y)$ over $y \geq \vec{0}$. Suppose $\hat{\pi}$ solves the dual of $L_{\hat{y}}$. Then $\hat{\pi}$ and $\hat{y}$ solve the dual of Q and the optimal value of Q is $v(\hat{y})$.

Intuition. Compared to Theorem 6.3.1, we now try to say something *backwards*. But we immediately see that it suffers from the situations depicts as follows.

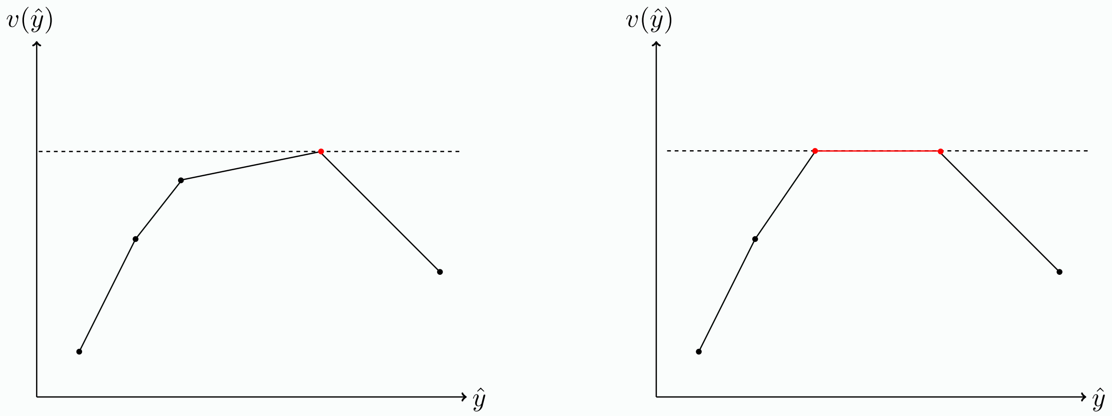


Figure 6.3: There may exist several $\hat{y}$!

Lecture 19: Lagrangian Relaxation

10 Nov. 08:00

As previously seen. We have

$$\begin{aligned} z := \min\ & c^\top x \\ & Ex \geq h \\ & Ax = b \\ (Q) \quad & x \geq 0 \end{aligned}$$

and by choosing $\hat{y} \geq \vec{0}$, we have the Lagrangian subproblem

$$\begin{aligned} v(\hat{y}) := \hat{y}^\top h + \min\ & (c^\top - \hat{y}^\top E)x \\ & Ax = b \\ (L_{\hat{y}}) \quad & x \geq 0. \end{aligned}$$

Now, we introduce another problem.

Definition 6.3.2 (Lagrangian dual). The *Lagrangian dual* problem is defined as

$$\max_{y \geq \vec{0}} v(y).$$

Note. We see that for $\hat{y} \geq \vec{0}$, $v(\hat{y}) \leq z$. Now, the goal is to solve the Lagrangian dual to get a lower bound for the original problem. (Notice that this is the maximum of the dual!)

Now, we try to proof Theorem 6.3.2, which is the *partial* converse of Theorem 6.3.1.

Proof of Theorem 6.3.2. Recall that

$$\begin{aligned}
v(\hat{y}) &:= \max_{y \geq \vec{0}} \; v(y) \\
&= \max_{y \geq \vec{0}} \left\{ y^\top h + \underbrace{\min_x \; \left\{ (c^\top - y^\top E)x \colon Ax = b, x \geq \vec{0} \right\}}_{\text{just a linear program}} \right\} \\
&= \max_{y \geq \vec{0}} \; \left\{ y^\top h + \max_{\Pi} \; \{ \Pi^\top b \colon \Pi^\top A \leq c^\top - y^\top E \} \right\} \\
&= \max_{y \geq \vec{0}, \Pi} \; \{ y^\top h + \Pi^\top b \colon \Pi^\top A + y^\top E \leq c^\top \} \\
&= z.
\end{aligned}$$

The last equality is derived from the fact that it's just the dual of the Q. ■

6.3.2 Solving the Lagrangian Dual

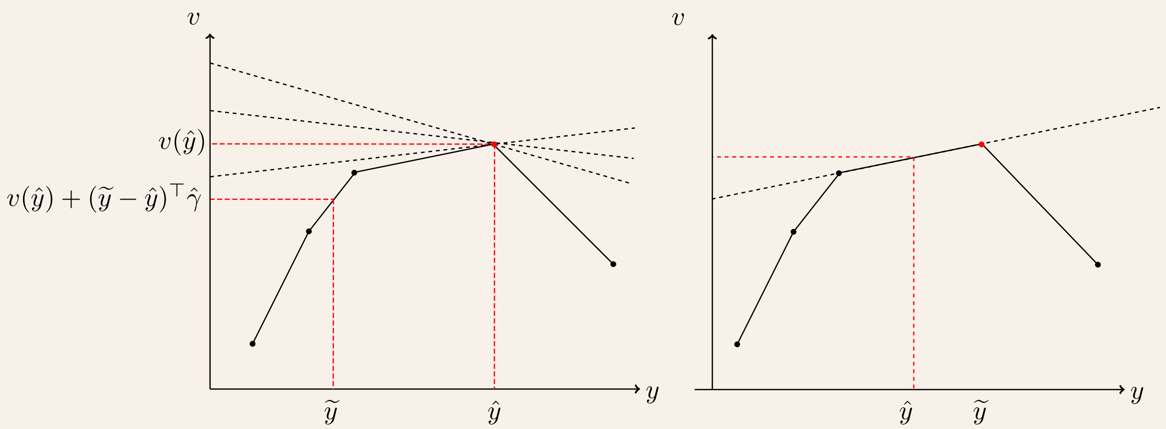
Intuition. Theorem 6.3.2 provides a simple way to calculate a lower bound on z by solving a potentially easier linear optimization problem. But we see that the bound depends on the choice of $\hat{y} \geq 0$. This push us to find the best such $\hat{y}$, and we indeed can solve this by solving the so-called Lagrangian dual problem of *maximizing* $v(y)$ over all $y \geq 0$ in the domain of v.

One may want to use some calculus technique to solve for such maximizing problem, but since v is not a smooth function, rather a piece-wise linear function, hence we need to introduce the concept of subgradient. Before we formally introduce it, we first see a theorem.

Theorem 6.3.3. Suppose we fix $\hat{y} \geq \vec{0}$ and solve for $v(\hat{y})$. Let $\hat{x}$ be the optimal solution of $L_{\hat{y}}$. Denote $\hat{\gamma} := h - E\hat{x}$, then

$$v(\widetilde{y}) \leq v(\hat{y}) + (\widetilde{y} - \hat{y})\hat{\gamma}$$

for all $\widetilde{y}$ in the domain of v.



Proof. We see that since

$$\begin{aligned}
v(\hat{y}) + (\hat{y} - \widetilde{y})\hat{\gamma} &= v(\hat{y}) + (\hat{y} - \widetilde{y})(h - E\hat{x}) \\
&= \hat{y}^\top h + (c^\top - \hat{y}^\top E)\hat{x} + (\widetilde{y} - \hat{y})^\top (h - E\hat{x}) \\
&= \widetilde{y}^\top h + (c^\top - \widetilde{y}^\top E)\hat{x} \\
&\geq v(\widetilde{y}).
\end{aligned}$$

The last inequality follows from the fact that $\hat{x}$ is only optimal for $L_{\hat{y}}$, not $L_{\widetilde{y}}$. $\hat{x}$ may just be feasible for $L_{\hat{y}}$. ■

In the theorem, $\hat{\gamma}$ is the so-called subgradient. Given $\widetilde{y}$ and $\hat{y}$, we choose $\hat{\gamma}$ such that the linear estimation $v(\hat{y}) + (\widetilde{y} - \hat{y})^\top \hat{\gamma}$ is always an upper bound on the value $v(\widetilde{y})$ of the function for all $\widetilde{y}$ in the domain of f. This $\hat{\gamma}$ is then a subgradient of (the concave function[1]) v at $\hat{y}$.

Mathematically, we have the following.

Definition 6.3.3 (Subgradient). For a concave function $f\colon I \to \mathbb{R}$, the *subgradient* (also known as *subderivative*) at point x_0 is a $c \in \mathbb{R}$ such that

$$f(x) - f(x_0) \geq c(x - x_0)$$

for every $x \in I$.

With this Theorem 6.3.3 about subgradient, we can then develop an algorithm to utilize this.

6.3.3 Projected Subgradient Optimization Algorithm

Intuition. We iteratively move in the direction of a subgradient to maximize v.

Algorithm 6.1: Projected subgradient optimization algorithm

Data: Objective function of Lagrangian dual $v(\hat{y}) = \hat{y}^\top h + \min(c^\top - \hat{y}^\top E)x$, maximum iteration K

Result: Estimated maximum value of $v(\hat{y})$

1 Initialize a random $\mathbb{R}^m$ vector $\hat{y}^\top \geq \vec{0}$
2 **for** $k = 1, \ldots K$ **do**
3   Solve $L_{\hat{y}^k}$ to get $\hat{x}^k$
4     $\hat{\gamma}^k \leftarrow h - E\hat{x}^k$
5     $\hat{y}^{k+1} \leftarrow \mathrm{Proj}_{\mathbb{R}^m_+}(\hat{y}^k + \lambda_k \hat{\gamma}^k)$
6 **return** $v(\hat{y}^K)$

Remark. There are a few remarks we want to make.

- The projection $\mathrm{Proj}_{\mathbb{R}^m_+}$ is just used to set any negative entries equal to 0.
- The key is in the line 5. We want to choose $\lambda_k > 0$ and satisfying something, which will make this algorithm converges.
  - **Harmonic step size**: Define the step size as $\lambda_k := \frac{1}{k}$, which will converge in theory, but it is slow. Notice that this choice of step size is *independent* of the current value of the subgradient.
  - **Polyak step size**: Define the step size as

    $$\lambda_k := \frac{\mathrm{GUESS} - v}{\|\hat{\gamma}^k\|^2},$$

    where we need an initial GUESS (we get this by literally *guessing*) to let the algorithm behaves reasonable.

Lecture 20: Convergence of Projected Subgradient Optimization Algorithm

As previously seen. We have already shown the algorithm of projected subgradient optimization, 17 Nov. 08:00
and the key is to choose an adequate step size λ_k. So we now try to give some conditions about how we can choose λ_k such that the algorithm converges.

[1] Contrast to Definition 5.2.1.

Lemma 6.3.2. Let y^* be any maximizer of v over $y \geq \vec{0}$. Suppose $\lambda_k > 0$ for all k. Then

$$\left\|y^* - \hat{y}^{k+1}\right\|^2 - \left\|y^* - \hat{y}^1\right\|^2 \leq \sum_{i=1}^{k} \lambda_i^2 \left\|\hat{\gamma}^i\right\|^2 - 2\sum_{i=1}^{k} \lambda_i \left(v(y^*) - v(\hat{y}^i)\right).$$

Proof. Let $w^{k+1} := \hat{y}^k + \lambda_k \hat{\gamma}^k$. Then for $k \geq 1$,

$$\begin{aligned}\left\|y^* - \hat{y}^{k+1}\right\|^2 - \left\|y^* - \hat{y}^k\right\|^2 &\leq \left\|y^* - w^{k+1}\right\|^2 - \left\|\hat{y} - \hat{y}^k\right\|^2 \\ &= \left\|(y^* - \hat{y}^k) - \lambda_k \hat{\gamma}^k\right\|^2 - \left\|y^* - \hat{y}^k\right\|^2 \\ &= \lambda_k^2 \left\|\hat{\gamma}^k\right\|^2 - 2\lambda_k (y^* - \hat{y}^k)^\top \hat{\gamma}^k \leq \lambda_k^2 \left\|\hat{\gamma}^k\right\|^2 - 2\lambda_k (v(y^*) - v(\hat{y}^k)),\end{aligned}$$

where the first inequality follows from the triangle inequality, and the last inequality follows from the definition of subgradient. We then do some *telescoping* and see that the result follows.

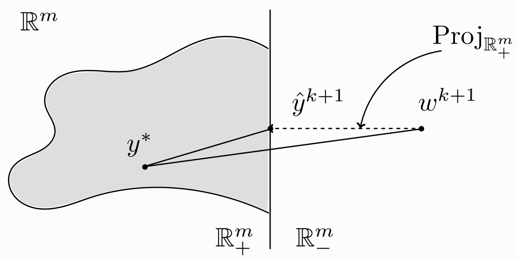


Figure 6.4: Triangle Inequality for $\mathrm{Proj}_{\mathbb{R}^m_+} w^{k+1} = \hat{y}^{k+1}$.

■

Now, denotes $v_k^* := \max_{i=1,\ldots,k} \left\{v(\hat{y}^i)\right\}$, which is just the best function value up to iteration k. Then we have the following result.

Theorem 6.3.4. Let y^* be any maximizer of v over $y \geq \vec{0}$. Assume that we take a basic optimal solution of $L_{\hat{y}^k}$. We further suppose $\lambda_k > 0$, $\sum_{k=1}^{\infty} \lambda_k = +\infty$, $\sum_{k=1}^{\infty} \lambda_k^2 < +\infty$. Then

$$\lim_{k\to\infty} v_k^* = v(y^*).$$

Proof. From the Lemma 6.3.2, the first term of the left-hand side is non-negative, hence we have

$$-\left\|y^* - \hat{y}^1\right\|^2 \leq \sum_{i=1}^{k} \lambda_i^2 \left\|\hat{\gamma}^i\right\|^2 - 2\sum_{i=1}^{k} \lambda_i \left(v(y^*) - v(\hat{y}^i)\right),$$

after rearrangement,

$$2\sum_{i=1}^{k} \lambda_i \left(v(y^*) - v(\hat{y}^i)\right) \leq \sum_{i=1}^{k} \lambda_i^2 \left\|\hat{\gamma}^i\right\|^2 + \left\|y^* - \hat{y}^1\right\|^2.$$

From the definition of v_k^*, we have

$$2\sum_{i=1}^{k} \lambda_i \left(v(y^*) - v_k^*\right) \leq \sum_{i=1}^{k} \lambda_i^2 \left\|\hat{\gamma}^i\right\|^2 + \left\|y^* - \hat{y}^1\right\|^2.$$

And since the $(v(y^*) - v_k^*)$ doesn't depend on i anymore, we can take it out of the summation. We further have

$$0 \le v(y^*) - v_k^* \le \frac{\sum_{i=1}^{k} \lambda_i^2 \left\|\hat{\gamma}^i\right\|^2 + \left\|y^* - \hat{y}^1\right\|^2}{2\sum_{i=1}^{k} \lambda_i}.$$

We observe that $\left\|y^* - \hat{y}^1\right\|^2$ is a constant, denotes it by c. Further, for all i, $\left\|\hat{\gamma}^i\right\|^2$ is bounded, so we can define

$$\Gamma := \max\left\{ \|h - Ex\|^2 : x \text{ is a basic feasible solution of } Ax = b,\, x \ge 0 \right\}.$$

With Γ, the inequality becomes

$$0 \le v(y^*) - v_k^* \le \frac{\Gamma \sum_{i=1}^{k} \lambda_i^2 + c}{2\sum_{i=1}^{k} \lambda_i} \to 0 \text{ as } k \to \infty$$

since we assume $\sum \lambda_i \to +\infty$ and $\sum \lambda_i^2 < +\infty$. Then we see $v_k^* = v(y^*)$ as $k \to \infty$ by squeeze theorem. ■

Remark. Suppose we instead choose

$$\lambda_k = s \in \mathbb{R}^+$$

being just a constant. Then the inequality in the above proof becomes

$$\frac{c + s^2 k\Gamma}{2ks} \to \frac{s\Gamma}{2}.$$

We see that with different choice λ_k, we can simply derive the upper-bound of $v(y^*) - v_k^*$.

6.4 Cutting-Stock Problem

So far we are talking about constraints being just positive, what about in other domain, like in $\mathbb{N}^+$?

Consider

$$\begin{aligned} \max \ & \sum_{i=1}^{n} x_i \\ & x_i + x_j \le 1, \text{ for all } 1 \le i < j \le n \\ & 0 \le x_i \le 1. \end{aligned}$$

This linear programming solution is $x_1 = x_2 = \cdots = x_n = \frac{1}{2}$ with the objective value being $\frac{n}{2}$. Denotes y as the dual variables. The dual is

$$\begin{aligned} \min \ & \sum_{i<j} y_{ij} \\ & \sum_{j\,:\, i\neq j} y_{ij} \ge 1 \text{ for all } i = 1, \ldots, n \\ & y_{ij} \ge 0 \end{aligned}$$

By setting $y_{ij} = \frac{1}{n-1}$, then the objective value is

$$\binom{n}{2} \frac{1}{n-1} = \frac{n}{2},$$

hence we confirm that $x_i = \frac{1}{2}$ is really the optimal solution. One can see that if now we let $x_i \in \mathbb{N}$, then the objective solution will be only one of $x_i = 1$, and the other $x_j = 0, j \neq i$. This leads to an optimal value being 1. This just shows how bad if we just **round down** the optimal solution when we consider so-called *integer programming*.

Lecture 21: Cutting-Stock Problem

It's now a good timing to introduce an application of what we have been discussing, namely the cutting-stock problem. We'll see that it naturally utilize the idea of column generation. It's an integer programming problem, though contrarily, it can be nicely approximated by **rounding down**. 22 Nov. 08:00

Problem 6.4.1 (Cutting-stock problem). Consider we have rolls of paper of width W, with the demand widths being $w_1, w_2, \dots, w_m < W$ and demands being (D), which is usually pretty big. The goal is to use as few stock rolls as possible.

Answer. One may try to define

$$x_{ij} := \#\text{ of rolls of width } w_i \text{ to cut from stock roll } j.$$

But we immediately see that the number of variables is huge for an integer programming, hence this doesn't work. Instead, we denote a *pattern* as a vector a being

$$a := \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix},$$

where $a_i = \#$ of pieces of width w_i to cut using this pattern. Then the constraints for a pattern is

$$\begin{aligned} &\sum_{i=1}^{m} w_i a_i \leq W \\ &\mathbb{N} \ni a_i \geq 0 \text{ for } i = 1, \dots, m. \end{aligned}$$

Moreover, denotes (D) as

$$d := \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix},$$

then we formulate the cutting-stock problem as

$$\begin{aligned} z := \min\ & \sum_j x_j \\ & \sum_j A_{\cdot j} x_j \geq d \\ \text{(CSP)}\quad & x_j \geq 0 \text{ integer for all } j, \end{aligned}$$

where

$$A_{\cdot j} = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix}.$$

Turning CSP into a standard form problem and **drop** the integer constraint, we have

$$\begin{aligned} \min\ & \sum_j x_j \\ & \sum_j A_{\cdot j} x_j - t = d \\ (\underline{\text{CSP}})\quad & x_j \geq 0 \text{ for all } j,\ t_i \geq 0 \text{ for all } i = 1, \dots, m. \end{aligned}$$

Note. $\underline{\text{CSP}}$ gives a lower bound on CSP. Moreover, the constraint of $x_j \in \mathbb{N}$ is now gone.

We now want to solve $\underline{\text{CSP}}$ exactly to get optimum $\overline{x}, \overline{t}$ with value $\underline{z} = \sum_{i=1}^{m} \overline{x}_i$.

Firstly, if we round up $\overline{x}$ to $\lceil \overline{x} \rceil$, then it is feasible for CSP. We immediately see

$$\sum_{i=1}^{m} \overline{x}_i = \underline{z} \leq z \leq \sum_{i=1}^{m} \lceil \overline{x}_i \rceil .$$

Since we also have

$$\left\lceil \sum_{i=1}^{m} \overline{x}_i \right\rceil \leq z,$$

hence we see that the rounding up solution $\lceil \overline{x} \rceil$ is within $m-1$ of optimum.

Now we consider how to solve $\underline{\text{CSP}}$ exactly. Denotes the dual variables of $\underline{\text{CSP}}$ being y such that

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}.$$

Suppose $\overline{y}$ is a basic dual solution. Then the reduced cost of a variable is:

- t_i:

$$0 - \overline{y}^{\top}(-e_i) = \overline{y}_i.$$

  Hence, if $\overline{y}_i < 0$, t can enter the basis.

- x_j:

$$1 - \overline{y}^{\top} A_{\cdot j} = 1 - \sum_{i=1}^{m} \overline{y}_i a_{ij}.$$

  If this is < 0, then x_j can enter the basis. To drive some quantity negative, we simply set up a minimization problem. Specifically, we set up a linear program such that

$$\begin{aligned} \min\ 1 - &\sum_{i=1}^{m} \overline{y}_i a_{ij} \\ &\sum_{i=1}^{m} w_i a_{ij} \leq W \\ &a_{ij} \geq 0 \text{ integers.} \end{aligned}$$

  Equivalently,

$$\begin{aligned} 1 - \max\ &\sum_{i=1}^{m} \overline{y}_i a_i \\ &\sum_{i=1}^{m} w_i a_i \leq W \\ &a_i \geq 0 \text{ integers for } i = 1, \dots, m. \end{aligned}$$

  This is known as the *Knapsack problem*. Now, let $f(S)$ being the optimal value for knapsack of capacity S such that $S = 0, 1, \ldots, W$. We see that

$$\begin{aligned} f(0) &= 0 \\ &\vdots \\ f(S) &= \max_{i\colon w_i \leq S} \{\overline{y}_i + f(S - w_i)\} \\ &\vdots \\ f(W) &= \text{ solution.} \end{aligned}$$

The running time is $\Theta(Wm)$.[a]

Notice that the above only gives $f(W)$, which is the objective value, but without information for variables. We can retrieve the information by keeping tracking of the argument of maximum in each step, namely we record

$$\begin{aligned} i_0^* \to f(0) &= 0 \\ &\vdots \\ i_S^* \to f(S) &= \max_{i\colon w_i \leq S} \{\overline{y}_i + f(S - w_i)\} \\ &\vdots \\ i_W^* \to f(W) &= \text{ solution.} \end{aligned}$$

Then we simply *back-track* every i^* from i_W^*, and then the next one is simply $i^*_{W - w_{i^*_W}}$, and so on.

⊛

[a]Note that we assume W and w_i to be integers.

Chapter 7

Integer-Linear Optimization

Lecture 22: Optimization of Integer Variables

Let first see some common pitfalls of integer programming. 29 Nov. 08:00

- If A has big entries and small entries, then these two constraints is like parallel to each other, which will lead the intersection be very far away. Then, if we simply round down the variable, the optimal value will drop significantly.

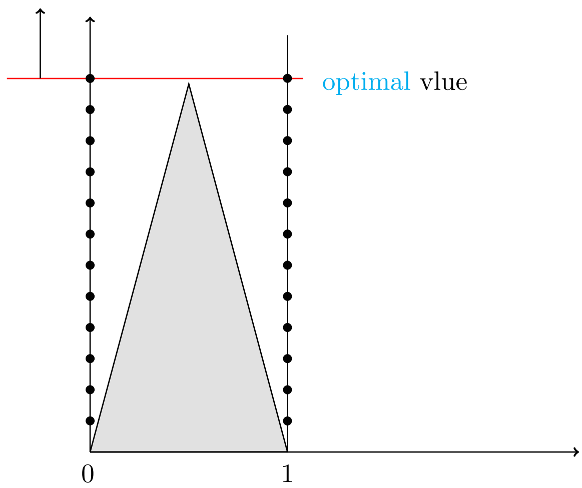


Figure 7.1: Pitfall of Integer Programming

But as one can see, we can often avoid this situation by carefully design our model and the problem is solved.

- Another possibility is the following. Consider an integer with the following constraints.

$$\begin{aligned}\forall_{1<i<j<n} \quad x_i + x_j &\leq 1 \\ \forall_{i=1,\ldots,n} \quad x_i \quad &\geq 0 \text{ integer.}\end{aligned}$$

Then, there are two feasible solutions one can observe immediately, namely

$$x_1 = x_2 = \cdots = x_n = \frac{1}{2};$$

and

$$x_1 = 1, x_2 = \cdots = x_n = 0.$$

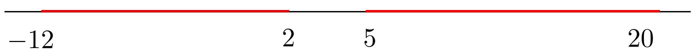
It's then really hard to tell which is better. But again, if the right-hand side is 2 for the first constraint, then the problem is gone.

Note. We see that this is totally opposite to the linear programming. The modern integer programming solver can easily solve a programming with like one hundred of variables, but the in practice, we're often facing more than thousands of variables. The one need to carefully design his model in terms of number of variables.

7.1 Modeling Techniques

We now introduce the so-called **Big-M Method**. Consider the following constraints.

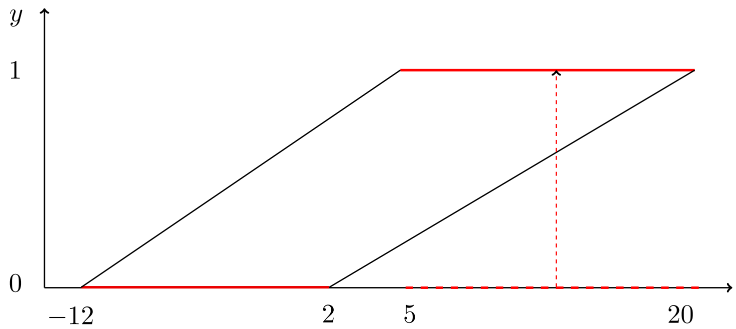
$$-12 \leq x \leq 2 \ \vee \ 5 \leq x \leq 20.$$



Then we need to find the smallest convex set which contains all feasible points. It's just

$$-12 \leq x \leq 20.$$

But that empty space between 2 and 5 causes the problem. To solve this, we simply introduce a new *indicator variable*, denotes it as y. y will be 0 if we are in $[-12, 2]$, and 1 if we are in $[5, 20]$. Then the smallest convex feasible region becomes the following quadrilateral.



Put it in the constraints, we have

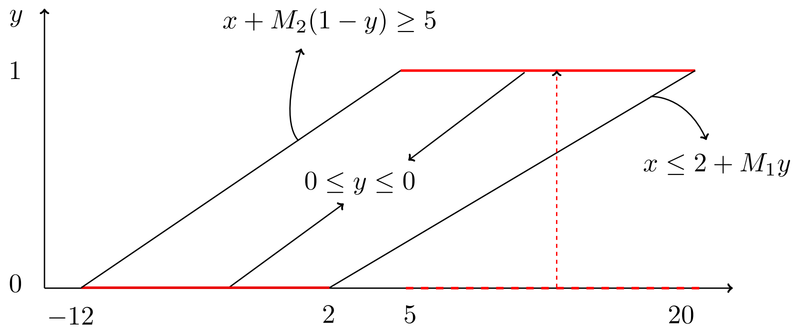
$$\begin{aligned} -12 \leq x &\leq 20 \\ 0 \leq y &\leq 1, \text{ integer} \\ x &\leq 2 + M_1 y \\ x + M_2(1-y) &\geq 5, \end{aligned}$$

where we let M_1 be big enough to let the constraints always be satisfied when x is in $[5, 20]$. For example, we can let $M_1 := 18$, then

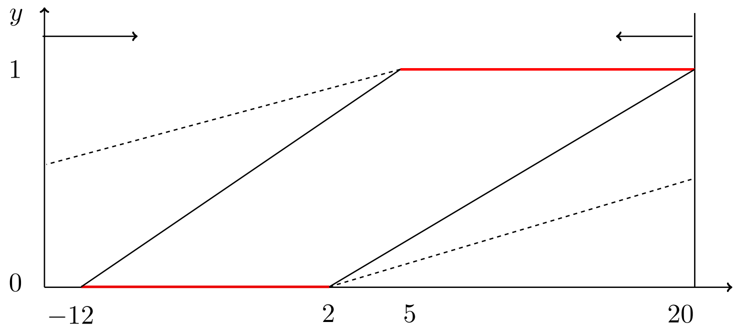
$$\begin{cases} y = 0, & x \leq 2 \\ y = 1, & x \leq 20. \end{cases}$$

Analogously, we use M_2 to help us to model the case that when $y = 1$, $x \geq 5$ and when $y = 0$, $x \geq -12$. For example, we can let $M_2 := 17$.

The last three constraints exactly corresponds to the line segment in the graph:



We further see that if we make the constant M_i too large, we will have



In terms of integer programming, this doesn't affect the integer feasible region. We call this **Big-M Method**. Although this is fine mathematically, but this is unfriendly to the solver.

7.1.1 Uncapacitated Facility-location Problem

Assume that there are m facilities with the fixed costs f_i, $i = 1, \ldots, m$. And assume there are n customers, denote by $j = 1, \ldots, n$. Now, let c_{ij} be the cost of satisfying all demand of customer j from facility i. The goal is to minimize the total cost of satisfying all customer's demand. We then define our variables as x_{ij} such that

$$x_{ij} \coloneqq \text{proportion of customer } j \text{ demand satisfied facility } i,$$

where $i = 1, \ldots, m$, $j = 1, \ldots, n$. Furthermore, we need indicator variables y_i such that

$$y_i \coloneqq \begin{cases} 1, & \text{if facility } i \text{ operates} \\ 0, & \text{if not} \end{cases}$$

for all $i = 1, \ldots, m$.

The optimization problem can now be modeled as

$$\begin{aligned} \min\ & \sum_{i=1}^{m} f_i y_i + \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} x_{ij} & \\ & \sum_{i=1}^{m} x_{ij} = 1, & \text{for } j = 1, \ldots, n \\ & -y_i + x_{ij} \leq 0, & \text{for } i = 1, \ldots, m,\ j = 1, \ldots, n \\ & 0 \leq y_i \leq 1 \text{ integers}, & \text{for } i = 1, \ldots, m \\ & x_{ij} \geq 0, & \text{for } i = 1, \ldots, m,\ j = 1, \ldots, n. \end{aligned}$$

Note. The third constraint

$$-y_i + x_{ij} \leq 0, \qquad \text{for } i = 1, \ldots, m,\ j = 1, \ldots, n$$

is for the following reason. For any i, if x_{ij} is positive for any j, then we need $y_i = 1$ to *force* us to pay the fixed cost to operate the facility if anything is shipped out of facility i. To get this constraint, we first see that we want

$$x_{ij} > 0 \Rightarrow y_i = 1$$

for any j. It is equivalent to say

$$\sum_{j=1}^{n} x_{ij} > 0 \Rightarrow y_i = 1.$$

We see that from the first expression, the constraint immediately follows. As for the second constraint, we start from considering

$$\sum_{j=1}^{n} x_{ij} \leq y_i$$

for every $i = 1, \ldots, m$. But this causes some problem. If the facility is really cheap, then the sum may exceed 1. To solve this problem, we simply make y become $n \cdot y$, namely

$$\sum_{j=1}^{n} x_{ij} \leq n \cdot y_i$$

for $i = 1, \ldots, m$, where n is just the **Big-M** in the big-M Method.

We now have two equivalent constraints, namely

$$\underset{i,j}{\forall} \; -y_i + x_{ij} \leq 0 \quad \text{and} \quad \underset{i}{\forall} \sum_{j=1}^{n} x_{ij} \leq n \cdot y_i.$$

Now the problem is which to use? The answer is the first one. We call the first model as the *strong model*, while the second model as the *weak model*.

Intuition. The second model has the big-M constant. As we just discuss, we prefer M to be as small as possible. But in the first model, we don't have that big-M coefficient. And since

$$\sum_{j=1}^{n} (x_{ij} \leq y_i) \Leftrightarrow \sum_{j=1}^{n} x_{ij} \leq n y_i,$$

we see that the weak constraint is just the sum over all strong constraint. In other words, we have

$$x_{ij} \leq y_i \Rightarrow \sum_{j=1}^{n} x_{ij} \leq n y_i.$$

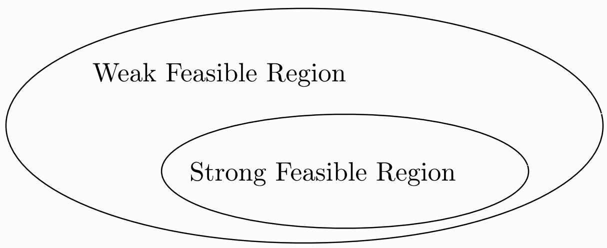
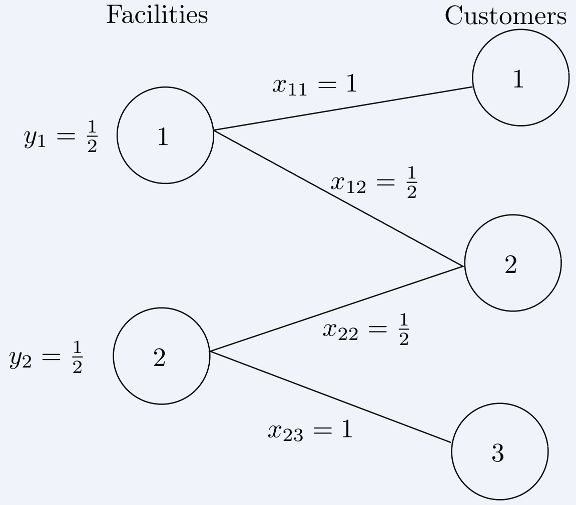


Figure 7.2: Venn diagram of strong and weak feasible region

Example. For $m = 2$, $n = 3$, find x, y where *weak* constraints are satisfied while *strong* constraints are not.



It's easy to check that $x_{11} \not\leq y_1$, but

$$x_{11} + x_{12} \leq 3y_1$$

and

$$x_{22} + x_{23} \leq 3y_2.$$

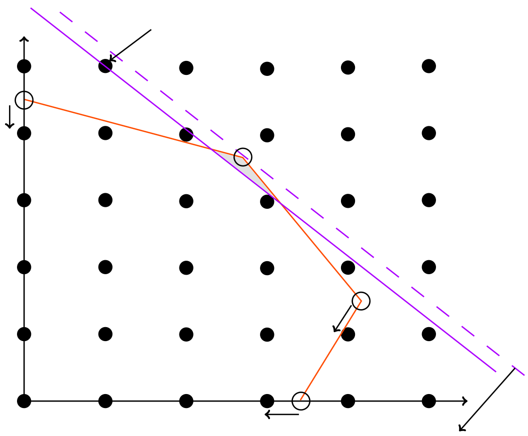
Remark. It's important to see that although we said we should keep the number of variables down when setting up the integer programming, but in this case, few is not always better!

Note (Disaggregation)**.** It's worth noting that the process of un-summing from the weak constraint to the strong constraint is called *disaggregation*.

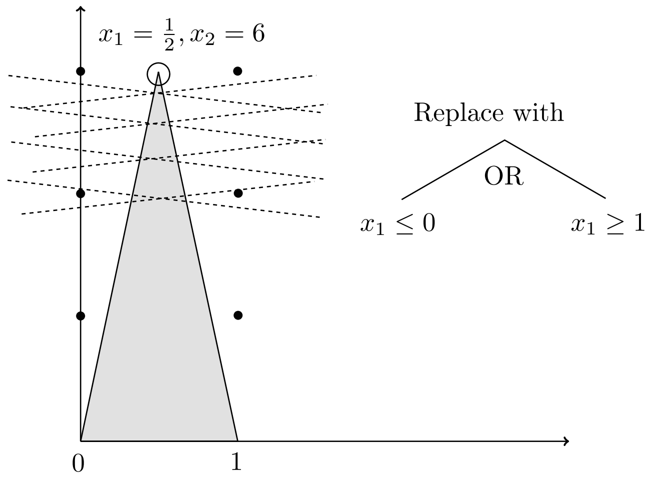
7.2 Algorithmically Solving Integer-Programming Problem

We now see some potential algorithm to solve the integer-programming problem.

- **Cutting-Plane** algorithm. If we have the following feasible region, then the *cutting-plane algorithm* suggests that we should use a plane at a corner (corresponds to an optimal solution to the linear version of this programming) and *reduce* the feasible region by a little until we touch an integer point.



- **Branch-and-Bound** algorithm. We first consider the following feasible region and try to use cutting-plane algorithm.



We see that if we simply start from cutting-plane algorithm, it takes forever to get to the answer. More generally, when the integer solution is far from the linear solution, the cutting-plane algorithm performs poorly.

Instead, we consider so-called *Branch-and-Bound algorithm*. It essentially just goes from n variables to two $n-1$ variables programming problem, and until we get to the bottom (1 variable). We see that we are doing exactly the opposite with what we have introduced, namely we are not modeling the **or**, but bring it into the algorithm. In this example, right after we branch, we solve the problem instantly since there in both branches, we only have one point to consider.

Note. Every modern solver which solves the integer programming exactly, will first go for branch-and-bound algorithm, and then on top of that, solve the remaining problem by cutting-plane algorithm.

Lecture 23: Branch and Bound Algorithm

7.2.1 Branch and Bound Algorithm

01 Dec. 08:00

We first dive into branch and bound algorithm as we mentioned.

As previously seen. The worst case in terms of time complexity for simplex algorithm is

$$\Theta(2^n - 1)$$

for n variables, but it's efficient in practice. And this is similar to the branch and bound algorithm for the integral programming problem.

We now focus on the following integer programming,

$$\begin{aligned} & \max \ y^\top b (=: z) \\ & \qquad y^\top A \leq c^\top \\ (\mathrm{D}_{\mathcal{I}}) \quad & \qquad y \in \mathbb{R}^m (y_i \in \mathbb{Z} \text{ for } i \in \mathcal{I}), \end{aligned}$$

where $\mathcal{I} \subseteq \{1, 2, \ldots, m\}$. By taking the dual, we have

$$\begin{aligned} & \min \ c^\top x \\ & \qquad Ax = b \\ (\mathrm{P}) \quad & \qquad x \geq 0. \end{aligned}$$

We'll see that the branch and bound algorithm maintains the following:

- $\mathcal{L}$: A list $\mathcal{L}$ of *subproblems* that have the form of $(\mathrm{D}_{\mathcal{I}'})$ where $\mathcal{I}' \subset \mathcal{I}$
- LB: The current best lower bound on z such that $\mathrm{LB} \leq z$.
- $\overline{y}_{\mathrm{LB}}$: The $\overline{y}$ corresponds to LB.

Note. LB is the objective value of the best feasible solution to the original problem seen so far. And we'll set

$$\mathrm{LB} \coloneqq -\infty$$

if there is no known feasible solution.

Remark. The key property of $\mathcal{L}$ is that if there is a feasible solution to the original problem that is better than LB, it should be feasible for some sub-problems on $\mathcal{L}$. Initially, we have

$$\mathcal{L} \coloneqq \{\mathrm{D}_{\mathcal{I}'}\} .$$

And we stop if

$$\mathcal{L} = \varnothing ,$$

since this implies $z = \mathrm{LB}$.

The general procedure is to take some problem $(\widetilde{\mathrm{D}}_{\mathcal{I}'})$ from $\mathcal{L}$ and remove it, and then solve its continuous relaxation $(\widetilde{\mathrm{D}})$ and proceed. Rigorously, we have the following pseudocode.

Algorithm 7.1: Branch and Bound Algorithm

Data: (Mixed) Integer programming $(\mathrm{D}_{\mathcal{I}})$ with *feasible* (P)[a]
Result: optimal solution $\overline{y}$ and optimal value, or report $(\mathrm{D}_{\mathcal{I}})$ is infeasible

$\mathcal{L} \leftarrow \{\mathrm{D}_{\mathcal{I}'} : \mathcal{I}' \subseteq \mathcal{I}\}$
$\mathrm{LB} \leftarrow -\infty$
$\overline{y}_{\mathrm{LB}} \leftarrow$ random vector

while $\mathcal{L} \neq \varnothing$ **do**
    $\mathrm{D}_{\mathcal{I}'} \leftarrow$ an element in $\mathcal{L}$ `// See subsection 7.2.3`
    $\mathcal{L} \leftarrow \mathcal{L} \setminus \{\mathrm{D}_{\mathcal{I}'}\}$
    result $\leftarrow$`SolveContRelax(`$\mathrm{D}_{\mathcal{I}'}$`)`
    if result *is not infeasible* **then**
        $\overline{y} \leftarrow$ result `// Retrieve basic optimal solution`
        if $\overline{y}^{\top} b > \mathrm{LB}$ **then**
            if $\overline{y}_i^{\top} \in \mathbb{Z}$ *for* $i \in \mathcal{I}$ **then** `//` $\overline{y}$ `solves` $\mathrm{D}_{\mathcal{I}}$
                $\mathrm{LB} \leftarrow \overline{y}^{\top} b$ `// Record optimal value`
                $\overline{y}_{\mathrm{LB}} \leftarrow \overline{y}$ `// Record optimal solution`
            else `//` $\overline{y}$ `doesn't solve` $\mathrm{D}_{\mathcal{I}}$, $\overline{y}_i^{\top} \notin \mathbb{Z}$ `for some` $i \in \mathcal{I}$
                $i^* \leftarrow$ an $i \in \mathcal{I}$ such that $\overline{y}_i \notin \mathbb{Z}$ `// See subsection 7.2.4`
                $\mathrm{D}^{\mathrm{u}}_{\mathcal{I}'} \leftarrow \mathrm{D}_{\mathcal{I}'}$ with $y_{i^*} \geq \lceil \overline{y}_{i^*} \rceil$ `// Up branch`
                $\mathrm{D}^{\mathrm{d}}_{\mathcal{I}'} \leftarrow \mathrm{D}_{\mathcal{I}'}$ with $y_{i^*} \leq \lfloor \overline{y}_{i^*} \rfloor$[b] `// Down branch`
                $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathrm{D}^{\mathrm{u}}_{\mathcal{I}'}, \mathrm{D}^{\mathrm{d}}_{\mathcal{I}'}\}$

if $\mathrm{LB} = -\infty$ **then**
    return *infeasible*
else
    return $\overline{y}_{\mathrm{LB}}$, LB

[a] This makes sure that the feasible region of the continuous relaxation $(\widetilde{\mathrm{D}})$ of any $(\widetilde{\mathrm{D}}_{\mathcal{I}'})$ is a bounded set, so we can guarantee finite termination.

[b] To match the form, we use $-y_i \leq -\lceil \overline{y}_i \rceil$.

Remark (Finite termination). We see that there are only finitely many (though exponential) $\mathrm{D}_{\mathcal{I}'}$ can be added into $\mathcal{L}$, hence branch and bound algorithm is finitely terminating.

Because the above algorithm maintains the key invariant for branch and bound, i.e., every feasible solution of $(\mathrm{D}_{\mathcal{I}})$ with greater objective value than LB is feasible for a problem on the list $\mathcal{L}$ we have the following result.

Theorem 7.2.1. Suppose that (P) is feasible. Then at termination of branch and bound, we have $\mathrm{LB} = -\infty$ if $(\widetilde{\mathrm{D}}_{\mathcal{I}})$ is infeasible, or with $\overline{y}_{\mathrm{LB}}$ being an optimal solution of $(\mathrm{D}_{\mathcal{I}})$.

Remark (Detail of the algorithm). We make some remarks on branch and bound algorithm.

- When calling line 8, we first obtain the continuous relaxation $(\widetilde{\mathrm{D}})$ of $(\mathrm{D}_{\mathcal{I}'})$ and its primal $(\widetilde{\mathrm{P}})$ as follows.

$$
\begin{array}{llll}
 & \max\ y^\top b & & \min\ c^\top x \\
 & \quad y^\top A \le c^\top & & \quad Ax = b. \\
(\widetilde{\mathrm{D}}) & & (\widetilde{\mathrm{P}}) & \quad x \ge 0
\end{array}
$$

  Then what we're really doing is solving $(\widetilde{\mathrm{P}})$ instead of $(\widetilde{\mathrm{D}})$ by simplex algorithm and get an optimal basis β, and this give us an optimal dual solution $\overline{y}^\top := c_\beta^\top A_\beta^{-1}$.

- When calling line 16, after choosing i^*, adding a constraint to $(\mathrm{D}_{\mathcal{I}'})$ effectively adds a variable to the corresponding continuous relaxation $(\widetilde{\mathrm{D}})$, hence adds a variable to the standard form problem $(\widetilde{\mathrm{P}})$. So, a basis for $(\widetilde{\mathrm{P}})$ remains feasible after we introduce such a variable.

  – Down branch: The constraint $y_{i^*} \le \lfloor \overline{y}_{i^*} \rfloor$ dualize to a new variable x_{down} in $(\widetilde{\mathrm{P}})$, which has a new column $A_{\mathrm{down}} := e_{i^*}$ and a cost coefficient $c_{\mathrm{down}} := \lfloor \overline{y}_{i^*} \rfloor$.

$$
\begin{array}{llll}
 & \max\ y^\top b & & \min\ c^\top x + \lfloor \overline{y}_{i^*} \rfloor x_{\mathrm{down}} \\
 & \quad y^\top A \le c^\top & & \quad Ax + e_{i^*} x_{\mathrm{down}} = b \\
(\widetilde{\mathrm{D}}) & \quad y_{i^*} \le \lfloor \overline{y}_{i^*} \rfloor & (\widetilde{\mathrm{P}}) & \quad x \ge 0, x_{\mathrm{down}} \ge 0.
\end{array}
$$

  The reduced cost of x_{down} is

$$
\overline{c}_{\mathrm{down}} = c_{\mathrm{down}} - \overline{y}^\top A_{\mathrm{down}} = \lfloor \overline{y}_{i^*} \rfloor - \overline{y}^\top e_{i^*} = \lfloor \overline{y}_{i^*} \rfloor - \overline{y}_{i^*} < 0
$$

  since $\overline{y}_{i^*}$ is not an integer. Hence, x_{down} is eligible to enter the basis.

  – Up branch: Similarly, we have a new variable x_{up} in $(\widetilde{\mathrm{P}})$ for the new constraint $y_{i^*} \ge \lceil \overline{y}_{i^*} \rceil$.

$$
\begin{array}{llll}
 & \max\ y^\top b & & \min\ c^\top x - \lceil \overline{y}_{i^*} \rceil x_{\mathrm{up}} \\
 & \quad y^\top A \le c^\top & & \quad Ax - e_{i^*} x_{\mathrm{up}} = b \\
(\widetilde{\mathrm{D}}) & \quad y_{i^*} \ge \lceil \overline{y}_{i^*} \rceil & (\widetilde{\mathrm{P}}) & \quad x \ge 0, x_{\mathrm{up}} \ge 0.
\end{array}
$$

  The reduced cost of x_{up} is

$$
-\lceil \overline{y}_{i^*} \rceil - \overline{y}^\top(-e_{i^*}) = \overline{y}_{i^*} - \lceil \overline{y}_{i^*} \rceil < 0,
$$

  since $\overline{y}_{i^*}$ is not an integer. Hence, x_{up} is eligible to enter the basis.

Remark (Partially solving $\widetilde{\mathrm{P}}$). In practice, when solving $(\widetilde{\mathrm{P}})$ (induced from any $(\mathrm{D}_{\mathcal{I}'})$) via simplex algorithm, we are generating a sequence of *decreasing objective values* of $(\widetilde{\mathrm{P}})$, each one of which is an upper-bound on the optimal value of its parent, and it's also a *potential new* LB. We see that when the optimal value of $(\widetilde{\mathrm{P}}) \le \mathrm{LB}$, we terminate immediately since solving this $(\widetilde{\mathrm{P}})$ will not improve LB.

7.2.2 Global Upper Bound

Since in practice, there are many errors in the data, so we may just want to solve it approximately, which means we only want to get a global upper bound. Conceptually,

$$\mathrm{UB} := \max\left\{\mathrm{LB}, \max\left\{\text{LP relaxation values for all problems on } \mathcal{L}\ \right\}\right\}$$

To calculate the set in the max, whenever children are created, solve their LP relaxation upon insertion into list. And we stop if

$$\mathrm{UB} - \mathrm{LB} < \text{absolute tolerance}.$$

> **Remark.** Apparently, we see that we can do this by reordering the algorithm. But for the original algorithm, we don't care about UB.

7.2.3 Node Selection

Node Selection means which problem to select from $\mathcal{L}$ to process. There are several ways to do this.

1. FIFO (First In First Out) $\cong$ Breadth First Search (BFS). New problems go at the end of the list, select from the front. We see that this strategy will **maximize memory usage**.
2. LIFO (Last In First Out) $\cong$ Depth First Search (DFS). New problems go to the first of the list, select from the front. We see that this strategy will **increase** LB **quickly**.
3. Best Bound. Need the LP upper bound for all problems on the list. We see that this strategy will **decrease** UB **quickly**.

> **Remark.** For any reasonable solver, it will first do the second strategy for several times, and they exclusively do the third strategy.

7.2.4 Branching Variable selection

1. Random: Choose randomly among y_i such that $\overline{y}_i \notin \mathbb{Z}$.
2. Biggest Cost: Choose based on the biggest c_i.
3. Most Fractional: Choose i with $\overline{y}_i$ *most fractional*.
4. **Pseudo Cost Branching**.

> **Note.** Someone argues that the *most fractional* rules is as bad as choosing randomly.

Lecture 24: Cutting Planes Algorithm

7.2.5 Cutting Planes Algorithm

06 Dec. 08:00

We focus on the problem in the form of

$$\begin{aligned} & \max\ y^\top b \\ & \quad\ y^\top A \leq c^\top \\ (\mathrm{D}_{\mathfrak{X}}) & \quad\ y_i \text{ integer for } i = 1, \ldots, m. \end{aligned}$$

> **Note.** Compare to subsection 7.2.1, now we require all y_i be integer, namely $\mathcal{I} = \{1, \ldots, m\}$ if we

refer to the used notation. Further, we also let (P) be

$$\begin{aligned} &\min\ c^\top x \\ &\quad Ax = b \\ (\text{P}) &\quad x \geq 0. \end{aligned}$$

Remark. We assume that the data A, c are all integer.

Now, we choose $w \in \mathbb{R}^n$, $w \geq 0$. Then the constraint in $(\mathrm{D}_{\mathfrak{X}})$ becomes

$$y^\top (Aw) \leq c^\top w.$$

Remark. This valid for all y such that

$$y^\top A \leq c^\top,$$

no matter it's integer or not.

Suppose $Aw \in \mathbb{Z}^m$. With the fact that $y \in \mathbb{Z}^m$, then for

$$y^\top (Aw) \leq c^\top w,$$

we can actually get

$$y^\top (Aw) \leq \left\lfloor c^\top w \right\rfloor$$

for all integer solutions of $y^\top A \leq c^\top$.

Definition 7.2.1 (Chvátal-Gomory cut). A *Chvátal-Gomory cut* is the inequality

$$y^\top (Aw) \leq \left\lfloor c^\top w \right\rfloor.$$

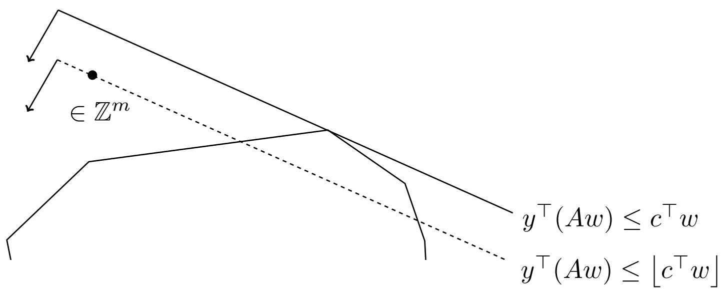


Figure 7.3: Chvatal-Gomory cut.

Remark. To get a Chvátal-Gomory cut, we need to choose $w \in \mathbb{Z}^n$, and we would like to develop a concrete algorithmic scheme for generating Chvátal-Gomory cuts. We'll do this via basic solutions.

We now solve (P) and get an optimal basis β. Consider the associated dual basic solution

$$\overline{y}^\top := c_\beta^\top A_\beta^{-1}$$

for the continuous relaxation of $(\mathrm{D}_{\mathfrak{X}})$.

Notice that if $\overline{y} \in \mathbb{Z}^m$, then $\overline{y}$ solves $(\mathrm{D}_{\mathfrak{X}})$. Otherwise, suppose $\overline{y}_i \notin \mathbb{Z}$, then let

$$\widetilde{b} := e_i + A_\beta r \in \mathbb{Z}^m,$$

where $r \in \mathbb{Z}^m$, our goal is to derive a valid cut for $(\mathrm{D}_{\mathfrak{X}})$ that is violated by $\overline{y}$.

Theorem 7.2.2. $\overline{y}^\top \widetilde{b} \notin \mathbb{Z}$, and so

$$y^\top \widetilde{b} \leq \lfloor \overline{y}^\top \widetilde{b} \rfloor$$

cuts off $\overline{y}$.

Proof. Since

$$\overline{y}^\top \widetilde{b} = \overline{y}^\top (e_i + A_\beta r) = \overline{y}_i + \overline{y}^\top A_\beta r = \overline{y}_i + c_\beta^\top A_\beta^{-1} A_\beta r = \overline{y}_i + c_\beta^\top r$$

We see that $\overline{y}_i \notin \mathbb{Z}$, $c_\beta^\top, r \in \mathbb{Z}$, hence we have

$$\overline{y}^\top \widetilde{b} = \overline{y}_i + c_\beta^\top r \notin \mathbb{Z}.$$

Now, we need to check that $y^\top \widetilde{b} \geq \lfloor \overline{y}^\top \widetilde{b} \rfloor$ is satisfied by $\overline{y}$.

Intuition. Consider if the inequality is

$$\vec{0}^\top y \leq -1,$$

then it makes no sense.

Hence, since $y^\top \widetilde{b} \geq \lfloor \overline{y}^\top \widetilde{b} \rfloor$, $y^\top \widetilde{b} \leq \lfloor \overline{y}^\top \widetilde{b} \rfloor$ indeed cuts off $\overline{y}$. ∎

We now need to show that the inequality $y^\top \widetilde{b} \leq \lfloor \overline{y}^\top \widetilde{b} \rfloor$ is valid for $(\mathrm{D}_{\mathfrak{X}})$. Let $H \coloneqq A_\beta^{-1}$, then $H_{\cdot i} = A_\beta^{-1} e_i$. Further, we let $w \coloneqq H_{\cdot i} + r$. Since we need $w \geq \vec{0}$, we can always choose $r \in \mathbb{Z}^m$ so that $w \geq \vec{0}$. Specifically, we choose

$$r_k \geq -\lfloor h_{ki} \rfloor$$

for $k = 1, \ldots, m$. Then, we have the following theorem.

Theorem 7.2.3. Choosing $r \in \mathbb{Z}^m$ satisfying $r_k \geq -\lfloor h_{ki} \rfloor$ for all k, we have

$$y^\top \widetilde{b} \leq \lfloor \overline{y}^\top \widetilde{b} \rfloor$$

is valid for $(\mathrm{D}_{\mathfrak{X}})$.

Proof. Because $y^\top A \leq c^\top$, we have $y^\top A_\beta \leq c_\beta^\top$. Then we have

$$\left(y^\top A_\beta\right) (H_{\cdot i} + r) \leq c_\beta^\top (H_{\cdot i} + r).$$

This is equivalence to

$$\left(y^\top A_\beta\right) \left(A_\beta^{-1} e_i + r\right) \leq c_\beta^\top \left(A_\beta^{-1} e_i + r\right).$$

After expanding, we have

$$y_i + y^\top A_\beta r \leq \overline{y}_i + c_\beta^\top r,$$

which can be written as

$$y^\top \underbrace{(e_i + A_\beta r)}_{\widetilde{b}} \leq \overline{y}^\top \underbrace{(e_i + A_\beta r)}_{\widetilde{b}}$$

since $\overline{y}^\top = c_\beta^\top A_\beta^{-1}$. Then, since $\widetilde{b} \in \mathbb{Z}^m$ and y is constrained to be in $\mathbb{Z}^m$ for $(\mathrm{D}_{\mathfrak{X}})$, we see

$$y^\top \widetilde{b} \leq \lfloor \overline{y}^\top \widetilde{b} \rfloor.$$

∎

Now, given any non-integer basic dual solution $\overline{y}$, we produce a valid inequality for $(\mathrm{D}_{\mathfrak{X}})$ from Theorem 7.2.2 that cuts it off.

Note. This cut for $(\mathrm{D}_{\mathfrak{X}})$ is used as a column for (P): the column is $\widetilde{b}$ with objective coefficient

$\lfloor \overline{y}^\top \widetilde{b} \rfloor$. Taking β to be an optimal basis for (P), the new variable corresponding to this column is the unique variable eligible to enter the basis in the context of the simplex algorithm applied to (P) since the reduced cost is precisely

$$\lfloor \overline{y}^\top \widetilde{b} \rfloor - \overline{y}^\top \widetilde{b} < 0.$$

Also, the new column for A is $\widetilde{b}$ which is integer, and the new objective coefficient for c is $\lfloor \overline{y}^\top \widetilde{b} \rfloor$, which is also an integer. So the original assumption that A and c are integer is maintained. Lastly, we need Aw are all integers. This is true since

$$A_\beta w = A_\beta \left(A_\beta^{-1} e_i + r \right) = e_i + A_\beta r \in \mathbb{Z}^m.$$

We can then repeatedly add new cuts.[a]

[a]Though we do our computations as column generation with respect to (P).

We note that there is clearly a lot of flexibility in how to choose r. The next (last!) theorem show that it's always best to choose a minimal $r \in \mathbb{Z}^m$ satisfying $r_k \geq -\lfloor h_{ki} \rfloor$ for all $k = 1, \ldots, m$.

Theorem 7.2.4. Let $r \in \mathbb{Z}^m$ be defined by

$$r_k = -\lfloor h_{ki} \rfloor, \text{ for } k = 1, \ldots, m.$$

Suppose that $\hat{r} \in \mathbb{Z}^m$ satisfies $\hat{r}_k \geq -\lfloor h_{ki} \rfloor$ for all $k = 1, \ldots, m$, then the cut determined by r dominates the cut determined by $\hat{r}$.

Proof. Obviously, $r \leq \hat{r}$. It's then easy to check that the cut can be re-expressed as

$$y_i \leq \lfloor \overline{y}_i \rfloor + (c_\beta^\top - y^\top A_\beta) r.$$

Noting that $c_\beta^\top - y^\top A_\beta \geq \vec{0}$ for all y that are feasible for the continuous relaxation of ($\mathrm{D}_{\mathfrak{X}}$), we see that the strongest inequality is obtained by choosing $r \in \mathbb{Z}^m$ to be minimal. ■

Revisiting the example. Now we see that the cutting plane algorithm will need at least $2k$ steps for such a triangle with height k, since it can only cut off one point at a time.

Example. Now we see some bad examples for Branch and Bound. Consider the following integer programming problem.

$$\begin{aligned} \min \ & y_{n+1} \\ & 2y_1 + 2y_2 + \cdots + 2y_n + y_{n+1} = \underbrace{n}_{\text{odd}} \\ & 0 \leq y_i \leq 1 \text{ for } i = 1, \ldots, n+1, \text{ integer}. \end{aligned}$$

We see that the optimum has $y_{n+1} = 1$.

If $n = 17$. Then we can let

$$y_{18} = 0, \quad y_1 = y_2 = \cdots = y_8 = 1, \quad y_9 = \frac{1}{2}, \quad y_{10} = \cdots = y_{17} = 0.$$

We immediately see there are lots of solutions like this, namely there are lots of symmetric groups going on such that half of the variables are 1, and another half of the variables are 0. This is pretty bad for the branch and bound algorithm since it will look at all of them. Analytically, we see that this will go into $\frac{n}{2}$ depth in the recursion tree, hence it's clearly exponential.

Remark (Pure v.s. Mixed). We only consider pure integer programming under the Gomory cutting plane scheme here, for mixed case, please refer to [Lee22].

Remark (Finite termination). To prove that Gomory cutting plane scheme is finitely terminating is rather technical in either pure or mixed case.[a]

We need to treat the objective function value as an additional variable (numbered first), employ the simplex algorithm adapted to the ϵ-perturbed problem, always choose the least-index $i \in \mathcal{I}$ having $\overline{y}_i \notin \mathbb{Z}$ and choose r via Theorem 7.2.4[b] as appropriate to generate the cuts.

[a]Though it's done in essentially the same manner.
[b]Or another rule for mixed case, see [Lee22]

Remark (Branch-and-Cut). SOTA algorithms for (mixed-)integer linear optimization (like `Gurobi`, `Cplex`) combine cuts with branch and bound, though it's too technical to make this work, so we'll skip it.

Bibliography

[Lee22] Jon Lee. *A First Course in Linear Optimization"*. Reex Press, 2022. URL: https://github.com/jon77lee/JLee_LinearOptimizationBook.