MATH561/IOE510/TO518
Linear Programming

Pingbang Hu

March 23, 2022

Abstract

We'll use [Lee22] as our main reference. This is a dynamic book which may changes and update constantly.

Contents

Lecture 1: Introduction 30 Aug. 08:00

1 Introduction to Linear Programming

1.1 General Linear Programming Problem

A general linear programming problem is to either minimize or maximize an *objective function* in the form of

$$c_1x_1 + c_2x_2 + \ldots + c_nx_n,$$

where x_i are our variables, $i = 1, \ldots, n$, and with the constraints

$$\begin{matrix} a_{11}x_1 + & \ldots & + a_{1n}x_n \gtreqless b_1 \\ a_{21}x_1 + & \ldots & + a_{2n}x_n \gtreqless b_2 \\ \vdots & \ddots & \vdots \\ a_{n1}x_1 + & \ldots & + a_{nn}x_n \gtreqless b_n \end{matrix},$$

which is sometimes called *structured constraints*, and finally with the constraints

$$x_1 \gtreqless 0, x_2 \gtreqless 0, \ldots, x_n \gtreqless 0,$$

which is called the *signed constraints*.

We called an assignment of values to variable x as a *solution*, and if this solution satisfies the linear constraints, we say that this solution is feasible. And a solution is *optimal* if there is no feasible solution with better objective value. Finally, the set of feasible solutions is called *feasible region*.

Remark. A feasible region is a polyhedron.

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

We will denote

$$c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}.$$

It's convenient to only consider so-called *Standard form problem*, which has the form of

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

with the condition that rows of A are linear independent, which means that no redundant equations and the system is consistent.

Remark. Notice that we only consider finitely many of constrains, since the property that the objective function can attain its extremum only on a compact set, which requires finite dimensional vector space.

Surprisingly, every linear programming problem can be converted to standard form, we now see how is this done.

1. Sign:
   - If $x_j \leq 0 \implies x_j \to -x_j^-$, where $x_j^- \geq 0$.
   - If x_j is unrestricted $\implies x_j \to x_j^+ - x_j^-$, where $x_j^\pm \geq 0$.
2. Constraints:
   - $\sum\limits_{j=1}^n a_{ij}x_j \leq b \implies \sum\limits_{j=1}^n a_{ij}x_j + s_i = b_i$, where $s_i \geq 0$. This s_i sometimes called *slack variable.*
   - $\sum\limits_{j=1}^n a_{ij}x_j \geq b \implies \sum\limits_{j=1}^n a_{ij}x_j - s_i = b_i$, where $s_i \geq 0$. This s_i sometimes called *surplus variable.*
3. Maximize: $\max \sum c_j x_j \implies -\min - \sum c_j x_j$.

Lecture 2: Duality

1 Sep. 08:00

1.2 First Glance of Duality

We can associate the standard form problem with another linear programming problem, called the *dual* of the original problem. We sometimes called the original problem the *primal.* The dual of the primal is

$$\begin{aligned} \min\ & c^T x & \qquad \max\ & y^T b \\ & Ax = b & & y^T A \leq c^T. \\ (P)\ & x \geq 0 & (D)\ & \end{aligned}$$

Note. We see that the dual is equivalent to

$$\begin{aligned} \max\ & b^T y \\ & A^T y \le c. \end{aligned}$$

Then we have a direct, but important theorem.

Theorem 1.1 (Weak duality theorem). If $\hat{x}$ is feasible for P, and $\hat{y}$ is feasible for D, then we have

$$c^T \hat{x} \ge \hat{y}^T b.$$

Proof. Since

$$\hat{y}^T A \le c^T \underset{\hat{x}\ge 0}{\Longrightarrow} \hat{y}^T A\hat{x} \le \hat{c}^T \hat{x} \underset{A\hat{x}=b}{\Longrightarrow} \hat{y}^T b \le c^T \hat{x}.$$

■

Example. Consider

$$\begin{aligned} \min\ & c^T x \\ & Ax \ge b, \end{aligned}$$

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

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

$$\begin{aligned} \min\ & c^T x \\ & Ax - S = b \\ & S \ge 0. \end{aligned}$$

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

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \qquad x^+ := \begin{pmatrix} x_1^+ \\ \vdots \\ x_n^+ \end{pmatrix}, \qquad x^- := \begin{pmatrix} x_1^- \\ \vdots \\ x_n^- \end{pmatrix}, x^\pm \ge \vec{0}.$$

Then we see the original problem becomes

$$\begin{aligned} \min\ & c^T (x^+ - x^-) \\ & A(x^+ - x^-) - S = b \\ & x^+, x^-, S \ge 0 \end{aligned}$$

We can further have

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

Set the dual variable being y, we further have

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

Note. The dual of the dual is the primal.

Lecture 3: Production Problem 8 Sep. 08:00

2 Production Problem

2.1 Production Problem

Skip.

add this!!!

2.2 Norm

Define

- maximum norm as
$$\|x\|_\infty := \max_{1 \leq i \leq n} \{|x_i|\}$$
- 1-norm as
$$\|x\|_1 := \sum_{i=1}^n |x_i|$$
- 2-norm as
$$\|x\|_2 := \sqrt{\sum_{i=1}^n x_i^2}$$

we can easily find the respective norm for x by following linear optimization problems.

2.2.1 Maximum(Infinity) Norm

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 the linear optimization *pressure* will force the maximum of $|x_i|$ being small, hence we'll get the minimum among $|x_i|$.

2.2.2 1-Norm

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 linear optimization pressure will force t_i goes to $|x_i|$, resulting $\sum\limits_{i=1}^{n} t_i$ being $\|x\|_1$.

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

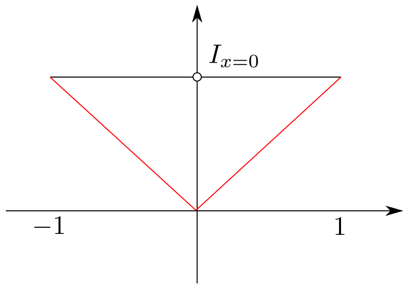


Figure 1: The best approximated convex function of $I_{x=0}$

Lecture 4: Basis Partition 13 Sep. 08:00

3 Algebra Versus Geometry

3.1 Elementary Row Operations

We can

1. permute rows
2. multiply a row by a non-zero factor
3. add a multiple of a row to another row
4. *permutation in columns*

What does 4. actually means? Consider

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

A permutation is a function like

$$(1, \ldots, n) \to (\sigma(1), \ldots, \sigma(n)).$$

Then the permuted matrix A_σ i

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

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,

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

3.2 Basic Partition

We denote a partition by

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

where β is called *basic*, while η is called *non-basic*. This is a partition of $\{1, \ldots, n\}$. The only condition we require for a basic partition is that

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

is invertible.

Associate a basic partition with a *basic solution* $\overline{x}$, which is **defined** as

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

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

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

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

Lecture 5: Convex Set

15 Sep. 08:00

3.3 Convex Set

Definition 3.1. A set $S \subseteq \mathbb{R}^n$ is a *convex set* if

$$x^1, x^2 \in S, \text{ and } 0 < \lambda < 1 \implies \lambda x^1 + (1-\lambda) x^2 \in S.$$

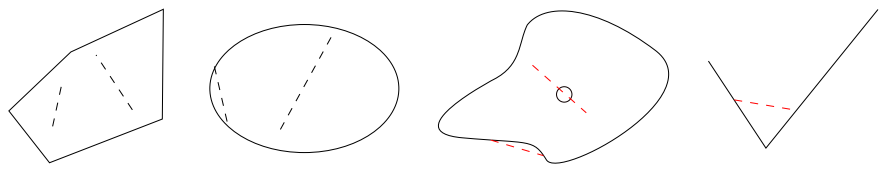


Figure 2: Convex Sets

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

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

Proof. Suppose there are two points x^1 and $x^2 \in S$ which means they are feasible. Consider a standard form problem, then we know

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

Then we simply have

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

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

3.4 Extreme Point

Definition 3.2. Suppose S is a convex set. Consider $\hat{x} \in S$. $\hat{x}$ is an *extreme-point* of S if we **cannot** write

$$\hat{x} = \lambda x^1 + (1-\lambda)x^2 \text{ with } x^1 \neq x^2, x^1, x^2 \in S, 0 < \lambda < 1.$$

Then we have an important theorem.

Theorem 3.1. 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

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

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

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

Theorem 3.2. If $\hat{x}$ is an extreme-point of the feasible region of P, then $\hat{x}$ is basic.

Proof. Skip... we leave it here. ■

Lecture 6: Feasible Direction and Ray

20 Sep. 08:00

3.5 Feasible Directions

We'll talk about an important concept, but before this, we first play around with the standard form problem a little. Consider

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

It's obvious that it's equivalent to

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

Further, we have

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

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

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

Note. $c_\eta^T - c_\beta^T A_\beta^{-1} A_\eta$ is what we called *reduced costs*. We'll see that we want this to be zero.

Now, with this intuition, we have the following definition.

Definition 3.3. Suppose $\hat{x} \in \mathcal{S}$, where $\mathcal{S}$ is a convex set. $\hat{z}$ is a *feasible direction* relative to $\hat{x}$ if

$$\exists \epsilon > 0, \quad \hat{x} + \epsilon \hat{z} \in \mathcal{S}.$$

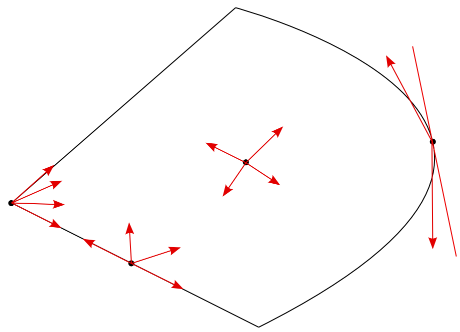


Figure 3: Feaisble Direction

We see that in order to let $\hat{z}$ to be a feasible direction, we need to have

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

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

Let the basic partition β, η being

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

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 basic and this j such that

$$\overline{z}_{\eta_j} = 1 \implies \overline{z}_\eta := e_j = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \leftarrow j$$

$$\overline{z}_\beta := -A_\beta^{-1} A_{\eta_j}.$$

Note. This needs

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

We check for feasibility:

1. $A\overline{z} = 0$: $A(\overline{x} + \epsilon \overline{z}) = b\surd$
2. $\overline{x} + \epsilon \overline{z}$:

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

Denote $\overline{b} := A_\beta^{-1} b$, $\overline{A}_{\eta_j} := A_\beta^{-1} A_{\eta_j}$, then

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

We finally have

$$\epsilon \leq \frac{\overline{b}_i}{\overline{a}_{i\eta_j}}, \qquad \underset{1 \leq i \leq m}{\forall} \overline{a}_{i\eta_j} > 0.$$

Notice that if $\overline{a}_{i\eta_j} \leq 0$, there is no restriction on ϵ being ≥ 0. Hence, we have a main result: $\overline{z}$ is a feasible direction from $\overline{x}$ if

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

Note. Notice that we can denote A by

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

Then since A_β is invertible, so

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

We now consider

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

We then have

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

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.

3.6 Feasible Rays

Definition 3.4. $\hat{z}$ is called a *ray* of a convex set $\mathcal{C}$ of $\hat{x} \in \mathcal{C}$ if

$$\forall \lambda > 0 \quad \hat{x} + \lambda \hat{z} \in \mathcal{C}.$$

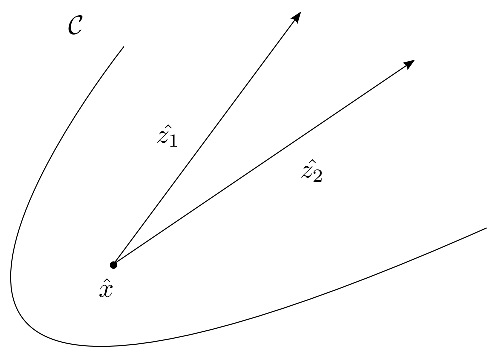


Figure 4: Ray

Suppose $\hat{x} \in \mathcal{C}$, where $\mathcal{C}$ 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

$$A(\hat{x} + \lambda\hat{z}) = \underbrace{A\hat{x}}_{=b} + \lambda \underbrace{A\hat{z}}_{=0} = b \implies \hat{z} \in n.s.(A).$$

Problem.

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

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

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

We have another concept about ray.

Definition 3.5. $\hat{z}$ is an *extreme ray* of a convex set $\mathcal{S}$ 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 $\mathcal{S}$ and $\mu \neq 0$.

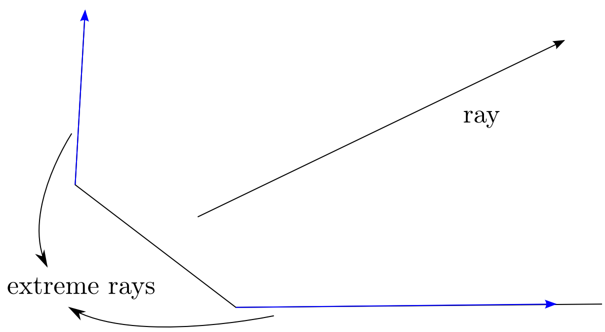


Figure 5: Extreme Rays

Summary. We can compare the *basic direction that are non-negative* with *extreme ray.*

$$\text{Basic solution } \overline{x} = \begin{cases} \overline{x}_\beta := A_\beta^{-1} b \geq 0 \\ \overline{x}_\eta := 0 \end{cases} \iff \text{Extreme points of the feasible region}$$

basic feasible direction	v.s.	**Geometry**
(Basic direction that are non-negative)		(Extreme Ray)

Lecture 7: Simplex Algorithm

22 Sep. 08:00

4 Simplex Algorithm

4.1 Simplex Algorithm

We start with considering the standard form problem

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

Definition 4.1. Define *dual basic solution* $\overline{y} \in \mathbb{R}^m$ as

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

Lemma 4.1. If β, η is a basic partition, and $\overline{x}, \overline{y}$ are the associated primal and dual basic solution, then

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

Proof.

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

■

As previously seen.

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

, and hence

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

We now formalize the concept of *reduced cost*.

Definition 4.2. $\overline{c}_\eta$ is called *reduced cost* for non-basic variables, where $\overline{c}_\eta$ is

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

4.1.1 Dual Feasibility

Lemma 4.2. $\overline{y}$ is feasible for D if and only if $\overline{c}_\eta \geq 0$.

Proof.

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

since

$$\begin{aligned} & y^T A_\beta \leq c_\beta^T \\ & y^T A_\eta \leq c_\eta^T \implies c_\eta^T - y^T A_\eta \geq 0. \end{aligned}$$

■

Corollary 4.1. If $\hat{x}$ is feasible for P and $\hat{y}$ is feasible for D, and if $c^T\hat{x} = \hat{y}^T b$, then $\hat{x}$ and $\hat{y}$ are optimal.

Theorem 4.1 (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

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

■

Note. The order of the arguments in text book for weak optimal basis theorem is slightly different.

4.1.2 Naive Algorithmic Approach

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

1 If $\overline{c}_\eta \geq 0$, then $\overline{x}$ and $\overline{y}$ are optimal and *STOP*.

2 Otherwise, choose η_j with $\overline{c}_{\eta_j} < 0$. Consider the associated basis direction $\overline{z}$.(Idea: $\overline{x} \to \overline{x} + \lambda\overline{z}$ with $\lambda > 0$) Then

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

where

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

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

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

3 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

$$\overline{A}_{i,\eta_j} \leq 0 \iff -A_\beta^{-1} A_{\eta_j} \geq 0 \iff \overline{z} \geq 0,$$

then $\overline{z}$ is a ray. This means P is unbounded below, hence we *STOP*.

4.1.3 Worry-Free Simplex Algorithm

Now, we give the very first generation about our simplex algorithm. Consider the standard form problem

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

The simplex algorithm is described as follows.

0. Start with a basic feasible partition β, η and assume that $x_\beta \geq 0$(x is a basic feasible solution)
1. (a) Compute $\overline{x}_\beta := A_\beta^{-1} b \geq 0$

   (b) Compute $\overline{c}_\eta^T := c_\eta^T - c_\beta^T A_\beta^{-1} A_\eta$

   If $\overline{c}_\eta \geq 0$, then *STOP*. $\overline{x}$ is optimal for P.
   (Recall that the dual solution $\overline{y} := c_\beta^T A_\beta^{-1}$ is optimal for D)
2. Otherwise, choose j such that $1 \leq j \leq n - m$ for η_j such that $\overline{c}_{\eta_j} < 0$. (Basic direction $\overline{z}$, then $c^T \overline{z} = \overline{c}_{\eta_j} < 0$)
3. Replace $\overline{x}$ with $\overline{x} + \lambda \overline{z}$ where

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

   (Largest choice so that $\overline{x} + \lambda \overline{z} \geq 0$)
   If we can't compute this, namely

   $$\overline{A}_{\eta_j} \leq 0 \implies P \text{ is unbounded } \implies \textit{STOP}$$

   Otherwise, *GOTO* 1.

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 basic index i^* will become non-basic, namely

$$(\overline{x} + \lambda \overline{z})_{\beta_{i^*}} = 0;$$

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

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

Namely

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

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

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

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 4.3. $A_{\widetilde{\beta}}$ is 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 $\det(A_\beta^{-1}A_{\widetilde{\beta}}) = \overline{a}_{i^*,\eta_j}$, if $\overline{a}_{i^*,\eta_j} \neq 0$, then we see that this is indeed invertible. But this is a obvious fact by our choice of i^*. ∎

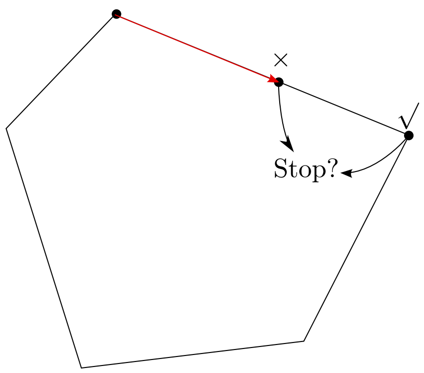


Figure 6: Pivot Swap in terms of feasible region.

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 4.4. 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 $x_{\widetilde{\eta}} = 0$ because $A_{\widetilde{\beta}}$ is invertible, namely

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

■

Lecture 8: Simplex Algorithm

27 Sep. 08:00

4.2 Remaining Problem

As previously seen. Simplex Algorithm:

1. Start with a basic feasible partition
   (a) i. Compute $\overline{x}_\beta := A_\beta^{-1} b \geq 0$
       ii. Compute $\overline{c}_\eta^T := c_\eta^T - c_\beta^T A_\beta^{-1} A_\eta$
   (b) If $\overline{c}_\eta \geq 0$, then *STOP*. $\overline{x}$ is optimal.
   (c) Otherwise, choose η_j such that $\overline{c}_{\eta_j} < 0$.
   (d) Define $i^* := \underset{i:\overline{a}_{i,\eta_j} > 0}{\arg\min} \{ \frac{\overline{x}_{\rho_i}}{\overline{a}_{i,\eta_j}} \}$
   (e) If i^* is undefined, then *STOP*. (P) is unbounded.
2. Swap β_i^* out of β and η_j out of η. *GOTO 1.*

Problem. How do we start with a basic feasible partition?

Answer. We consider the so-called *Phase One Problem.*

4.2.1 Phase one problem

$$\begin{aligned} \min\ & c^T x \\ & Ax = b \\ & x \geq 0 \end{aligned} \qquad (P)$$

$$\begin{aligned} \min\ & x_{n+1} \\ & Ax + A_{n+1} x_{n+1} = b \\ & x \geq 0, x_{n+1} \geq 0 \end{aligned} \qquad (\Phi)$$

1. If min value of x_{n+1} in Φ is 0, then we get a feasible solution of (P).

2. If min value of x_{n+1} in Φ is > 0, then there is no feasible solution of (P).
   - How do we get an initial basic feasible solution for Φ
   - Need a basic feasible solution.

Solution:

1. Start with a basic solution of (P), $\tilde{\beta}, \tilde{\eta}$ is the basic partition.
2. If its feasible($\overline{x}_{\tilde{\beta}}$) then we just use $\tilde{\beta}$ and $\tilde{\eta}$ for β and η
3. 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

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

Problem. What if $x_{n+1} = 0$?

Intuition. Just stop right before $x_{n+1} = 0$, let other variable do that.

4.2.2 Non degeneracy hypothesis

$$\begin{aligned} & \vec{x}_{\beta_i} > 0 \text{ for all } i \text{ at every iteration} \\ \implies & \overline{\lambda} \neq 0 \\ \implies & \text{objective value decrease at each iteration.} \\ \implies & \text{algorithm must (because a finite \# of basis)} \end{aligned}$$

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

where ϵ is an arbitrarily small *indeterminate*.

Remark.

$$\epsilon \neq 0.$$

Observe. polynomial in ϵ:

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

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

Definition 4.3. 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.

$$p(\epsilon) = p_0 + p_1\epsilon + p_2\epsilon^2 + \cdots + p_m\epsilon^m.$$
$$q(\epsilon) = q_0 + q_1\epsilon + q_2\epsilon^2 + \cdots + q_m\epsilon^m.$$

with K_p and K_q. Then K_{p+q} depends on K_p and K_q.

$$p(\epsilon) - q(\epsilon) \geq 0? \qquad \text{Then } p(\epsilon) \geq q(\epsilon).$$

Problem. Where does this ϵ thing links with the Simplex algorithm? (d)

4.2.3 Perturbed Problem

Suppose

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

Feasible for perturbed problem means $p(\epsilon) \geq \vec{0} \implies p(0) = p_0 \geq 0$.

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

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

$$\vec{x}_\beta = A_\beta^{-1}(b + A_\beta \vec{\epsilon}) = \underbrace{A_\beta^{-1} b}_{\geq \vec{0}} + \vec{\epsilon} = \vec{x}_\beta + \begin{pmatrix} \epsilon \\ \epsilon^2 \\ \epsilon^3 \\ \vdots \\ \epsilon^4 \end{pmatrix} = \begin{pmatrix} \vec{x}_{\beta_1} + \epsilon \\ \vec{x}_{\beta_2} + \epsilon^2 \\ \vdots \\ \vec{x}_{\beta_m} + \epsilon^m \end{pmatrix} \geq \vec{0}.$$

Claim: Perturbed problem is non-degenerate. $\implies$ some later basis $\tilde{\beta}$

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

$(\vec{x}_{\tilde{\beta}_i} \overset{?}{=} 0)$

$$i^{\text{th}} \text{ element of } A_{\tilde{\beta}}^{-1} A_\beta \begin{pmatrix} \epsilon \\ \epsilon^2 \\ \epsilon^3 \\ \vdots \\ \epsilon^m \end{pmatrix} \implies \underbrace{i^{\text{th}} \text{ row of } A_{\tilde{\beta}}^{-1} A_\beta}_{= \vec{0}} \text{ dot } \vec{\epsilon} \; \lightning$$

because $A_{\tilde{\beta}}^{-1} A_\beta$ is invertible($A_\beta^{-1} A_{\tilde{\beta}}$ is the inverse)

Lecture 9: Practical Simplex Algorithm

29 Sep. 08:00

4.2.4 A_β^{-1} in Reality

Note. In reality, we don't really calculate A_β^{-1}, since in order to calculate

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

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.

4.3 Why *Simplex*?

For a standard form problem

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

But instead, we consider

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

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

4.3.1 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

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

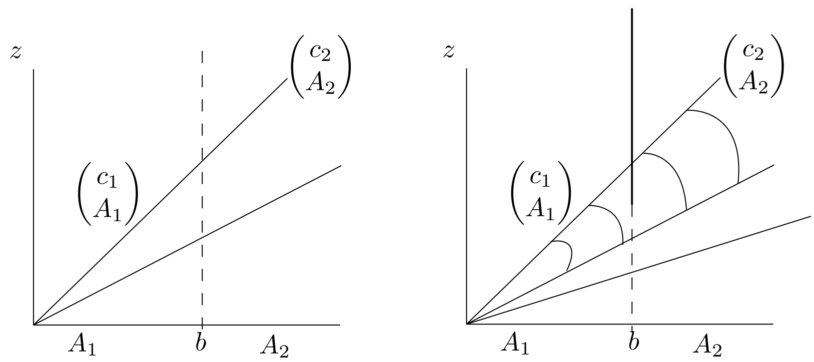


Figure 7: column-geometry

4.3.2 Simplices(plural of simplex)

Figure 8: Simplex shapes

Example. Example of a simplex:

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

Which is $n-1$ dimensional simplex in $\mathbb{R}^n$ with n standard unit vectors are the corners.

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

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

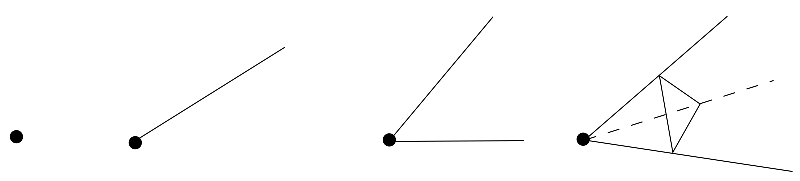


Figure 9: Simplicial Cones

4.4 Complete Simplex Algorithm

Now, we have the standard form problem (P):

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

and the *Phase one problem*(Φ)(Getting started): $A_{n+1} = -A_{\tilde{\beta}}\vec{1}$:

- First pivot is special
- Last pivot is special

:

$$\begin{aligned} \min\ & x_{n+1} \\ & Ax + A_{n+1}x_{n+1} = b \\ & x \geq 0, x_{n+1} \geq 0 \end{aligned} \tag{Φ}$$

with the perturbed problem (P_ϵ)(making sure we stop):

$$\begin{aligned} \min\ & c^T x \\ & Ax = b + B\vec{\epsilon} \\ & x \geq 0 \end{aligned} \tag{P_ϵ}$$

5 Duality

Consider the standard problem and its duality:

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

As previously seen. Weak duality theorem: If $\hat{x}$ is feasible for P, and $\hat{y}$ is feasible for D, then

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

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

Theorem 5.1 (Weak optimal basis theorem). 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)

$$\implies \overline{x} \ \&\ \overline{y} \text{ are optimal.}$$

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

Theorem 5.2 (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

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

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 5.3 (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

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

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 I problem and the perturbation) is important.

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

Table 1: Comparison between P and D

Lecture 10: Duality

4 Oct. 08:00

5.1 Complementary

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

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

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

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

Note. Specifically, we see that $c_j - \hat{y}^T A_{\cdot j} = 0$ for $j \in \beta$ is because $\overline{y}^T = c_\beta^T A_\beta^{-1}$, and then

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

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

Theorem 5.4. If $\overline{x}$ and $\overline{y}$ are basic solutions for β, η, then $\overline{x}$ and $\overline{y}$ are complementary.

Theorem 5.5 (Complementary with equal objective value). If $\hat{x}$ and $\hat{y}$ are complementary, then

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

Note.

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

Proof. We show that

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

We have

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

■

Theorem 5.6 (Weak complementary slackness theorem). If $\hat{x}$ and $\hat{y}$ are feasible and complementary, then they are optimal.

Proof. Follows from weak duality theorem(Theorem 1.1) and complementary solutions having equal objective value(Theorem 5.5). ■

Theorem 5.7 (Strong complementary slackness theorem). If $\hat{x}$ and $\hat{y}$ are optimal, then $\hat{x}$ and $\hat{y}$ are complementary.

Proof. Recall that

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

Hence, the equality can only hold if

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

with the obvious fact that

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

so they are complementary. ■

5.2 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

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

We first turn this into a standard form problem:

1. $\widetilde{x}_N := -x_N$:

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

2. $x_U = \widetilde{x}_U - \widetilde{\widetilde{x}}_U$, where $\widetilde{x}_U, \widetilde{\widetilde{x}}_U \geq 0$:

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

3. Adding slack variables:

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

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

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

We time -1 to the both side of the second constraint, and we see that last two structure constraints can be reduced to a single equality, results in

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

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

Theorem 5.8. We rephrase the weak and strong duality theorem in a more general term.

- 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

$$c_P^T \hat{x}_P + c_N^T \hat{x}_N + c_U^T \hat{x}_U \geq \hat{y}_G^T b_G + \hat{y}_L^T b_L + \hat{y}_E^T b_E.$$

- 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,

$$c_P^T \hat{x}_P + c_N^T \hat{x}_N + c_U^T \hat{x}_U = \hat{y}_G^T b_G + \hat{y}_L^T b_L + \hat{y}_E^T b_E.$$

Remark. We can also rephrase the Weak Complementary Slackness Theorem (Theorem 5.6) and also the Strong Complementary Slackness Theorem (Theorem 5.7) in this setup. The proof follows the same idea, but with some more works.

Lecture 11: Duality

6 Oct. 08:00

As previously seen. Complementary: we have

$$\begin{aligned}\min\ & c^T x & \qquad \max\ & y^T b \\ & Ax = b & & y^T A \leq c^T. \\ (P)\quad & x \geq 0 & (D) & \end{aligned}$$

Then the complementary means that

$$\underbrace{(c_j - \hat{y}^T A_{\cdot j})}_{\geq 0} \underbrace{\hat{x}_j}_{\geq 0} = 0 \text{ for } j = 1 \dots n$$
$$\hat{y}_i \underbrace{(A_{i\cdot}\hat{x} - b_i)}_{=0} = 0 \text{ for } i = 1 \dots m.$$

As previously seen. The production problem: The primal:

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

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

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

where

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

Note. We have

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

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

Come back to complementary.

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

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

Since

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

Observe. We want

$$c^T \leq y^T A \implies c^T x \leq y^T A x$$

Now, return to the standard form problem, we have

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

with y unrestricted.

Then we have

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

since

$$y^T A x \leq c^T x.$$

Example. Consider the following linear programming problem:

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

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

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

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

$$\begin{aligned} & y^T A + w^T B \leq c^T \\ & (y^T A + w^T B) x \geq c^T x\text{'} \end{aligned}$$

hence

$$\begin{aligned} & \overbrace{y^T}^{\leq 0} (Ax \geq b) \\ + \ & w^T (Bx - Fz = g) \end{aligned}$$

$$c^T x + d^T z \overset{\text{want}}{\leq} \underbrace{y^T A x + w^T B x - w^T F z}_{\underbrace{(y^T A + w^T B)}_{\leq c^T} \underbrace{x}_{\leq 0} \underbrace{-(w^T F) z}_{= d^T}} \overset{\text{want}}{\leq} y^T b + w^T g$$

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

Rethink about it

5.3 Geometrically Understanding of Duality

5.3.1 Farkas' Lemma

Lemma 5.1. Farkas' Lemma: Let (I) and (II) being

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

for any data A and b, exactly one of (I) or (II) has a solution.

Note. Recall that the *LP Duality*

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

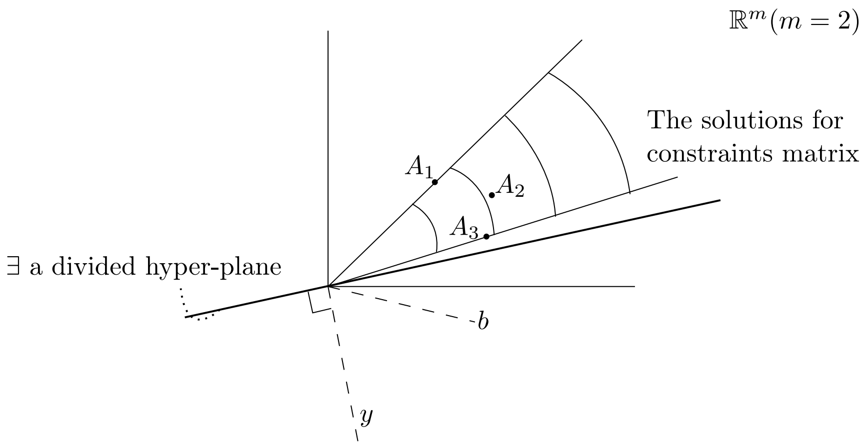


Figure 10: Farkas Lemma - Geometrically point of view with $\mathbb{R}^m, m = 2$

Intuition. We outline the idea about the proof.

1. 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

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

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

Lecture 12: Farkas' Lemma

11 Oct. 08:00

Before we prove Farkas' Lemma, we first see something similar. There is a lemma called *Gauss' Lemma*, which is highly related to Farkas' Lemma.

Lemma 5.2. Gauss' Lemma : Exactly one of the following has a solution :

$$
\begin{aligned}
(I) \qquad & Ax = b \\
\\
(II) \qquad & y^T A \geq 0 \\
& y^T b \neq 0
\end{aligned}
$$

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

1. The system has no solution.
2. 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 we start to proof Farkas' Lemma.

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

1. I & II can't both have solutions. Suppose $\hat{x}$ solves I and $\hat{y}$ solves II. Then we have
$$\hat{y}^T(\hat{Ax} = b) \implies \underbrace{(\hat{y}^T A)}_{\geq \vec{0}} \underbrace{\hat{x}}_{\geq \vec{0}} = \hat{y}b > 0 \lightning$$

2. At least one of I or II has a solution $\cong$ If I has no solution, then II has a solution. Assume that I has no solution, which means that P is infeasible with P being
$$
\begin{aligned}
& \min \ \vec{0}^T x \\
& \qquad Ax = b \\
(P) \quad & \qquad x \geq 0.
\end{aligned}
$$

The dual of this P is

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

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

$$\begin{aligned} &\implies D \text{ is unbounded} \\ &\implies \text{ there exist a feasible solution } \widetilde{y} \text{ to } D \text{ with positive objective} \end{aligned}$$

■

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

$$\begin{aligned} &\min\ c^T x \\ &\qquad Ax = b \\ (P)\quad & x \ge 0 && \text{feasibility} \\ & && \Updownarrow \\ &\max\ y^T b && \text{unbounded direction} \\ (D)\quad & y^T A \le c^T \end{aligned}$$

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

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

Furthermore, we have

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

Example.

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

Find out what II is.

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

$$\begin{aligned} &\min\ \vec{0}^T x && \max\ y^T b \\ &\qquad Ax \le b && \qquad y^T A = \vec{0}. \\ (P) & && (D)\quad y \le \vec{0} \end{aligned}$$

Then we have

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

Check:

$$0 = \underbrace{\hat{y}^T A}_{=\vec{0}} \hat{x} \underset{\hat{y} \le \vec{0}}{\ge} \hat{y}^T b > 0 \lightning$$

or,

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

Example.

$$\begin{aligned} & (\min\ \vec{0}^T x + \vec{0}^T w) \\ & \quad A\ x + B\ w = b \\ & \qquad\ - F\ w \ge f \\ (I) & \qquad x \ge 0, w \text{ unrestricted} \end{aligned}$$

with the dual variables y, w, we have

$$\begin{aligned} & \text{(Suppose I has no solution.)} \\ \cancel{\max}\ & y^T b + v^T b (> 0) \\ & y^T A \le \vec{0} \\ (II) \quad & y^T B - v^T F = \vec{0} \end{aligned}$$

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

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

$$\begin{aligned} (I) \quad & Ax = b \\ & x \ge 0 && \iff b \text{ is in the cone } K \\ (II) \quad & y^T b > 0 && \iff y \text{ makes an acute angle with } b. \\ & y^T A \le 0^T && \quad y \text{ makes a non-acute angle with all columns of } A \end{aligned}$$

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

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

Then we have

$$y^T \hat{z} = \underbrace{y^T A}_{\le \vec{0}^T} \underbrace{\vec{x}}_{\ge \vec{0}} \le 0.$$

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

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

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

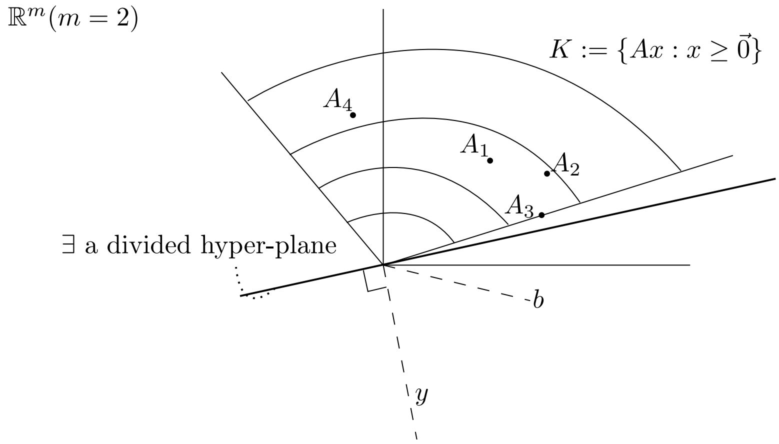


Figure 11: Case II of the Farkas' Lemma with $m = 2$

5.4 The Big Picture of Cones

Consider the linear programming problem

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

with the partition β, η, we see that

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

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

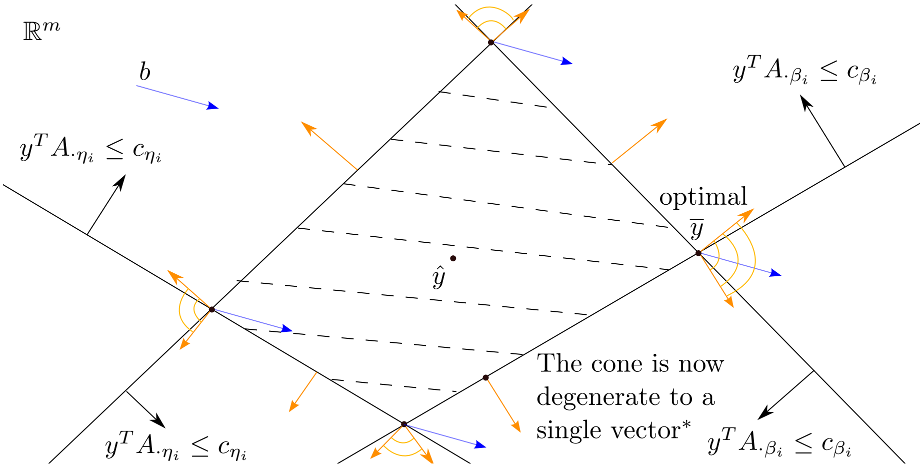


Figure 12: Optimality of Cones. (* This corresponds to the case that we run into the overlapping issue in Figure 13)

with

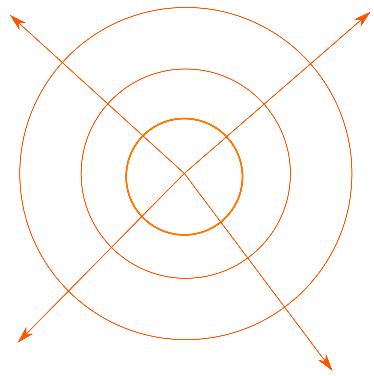


Figure 13: Cones join together

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

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

Example. Exercise 5.5. ~~Over~~ Strictly Complementarity. Consider

$$
\begin{array}{llcll}
\min & c^T x & \qquad & \max & y^T b \\
 & Ax = b & & & y^T A \leq \vec{0}. \\
(P) & x \geq 0 & & (D) &
\end{array}
$$

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

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

Definition 5.1. For feasible solutions $\hat{x}$ and $\hat{y}$ are strictly complementary if they are complementary and exactly one of

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

Theorem 5.9 (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:

$$
\begin{array}{rl}
v = \min & c^T x \\
 & Ax = b \\
(P) & x \geq 0
\end{array}
$$

Now, we try to find an optimal solution with

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

by formulating the following linear programming

$$
\begin{array}{rl}
\max & x_j \\
 & c^T x \leq v \\
 & Ax = b \\
(P_j) & x \geq 0
\end{array}
$$

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

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

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

18 Oct. 08:00

As previously seen. Strictly complementary.

Proof. First prove for one fixed j. Consider

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

where

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

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.

I. P_j has an optimal solution. $\hat{x}$ with $\hat{x}_j > 0$. Take $\hat{x}$ optimal for $P_j \implies \hat{x}$ optimal for P. Take an $\hat{y}$ optimal for D.

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

III. 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

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

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

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

$$\begin{aligned} & -c^T + \left(\frac{\hat{y}^T}{-\hat{w}}\right) A \underset{\leq}{\cancel{\geq}} \frac{1}{-\hat{w}} e_j^T \\ \implies & \underbrace{\left(\frac{\hat{y}^T}{-\hat{w}}\right)}_{\hat{\hat{y}}} A \leq c^T - \frac{1}{\hat{w}} e_j^T \\ \implies & \hat{\hat{y}}^T A \leq c^T - \frac{1}{\hat{w}_j} e_j^T \\ \implies & \hat{\hat{y}}^T A \leq c^T \text{ with a little slack in the } j^{th} \text{ constraint.} \\ \implies & \hat{\hat{y}}^T A_{\cdot j} \leq c_j - \frac{1}{\hat{w}} < c_j, \forall j. \end{aligned}$$

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

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

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

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

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

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

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

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

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

■

Notice that this is just for a fixed j!

j		$\hat{x}^T$					$c^T - \hat{y}^T A$	
1	$\ddots$		0		$\ddots$			
$\vdots$		$\ddots$	$\vdots$			$\ddots$		
j	$\to \hat{x}^{(j)}$	0/+	$\vdots$				+/0	$\leftarrow c^T - \hat{y}^{(j)T} A$
$\vdots$			$\ddots$				$\ddots$	
n			0	$\ddots$			+	$\ddots$
		$\hat{\hat{x}}$	$\uparrow$				$c^T - \hat{\hat{y}}^T A$	
			0				+	

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

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

Problem. For multicommodity flow problem, we see that

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

Answer. Write it in the matrix form, we have

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

write it in another way, we have

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

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

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

The A looks like

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

Then we see the dual is just

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

Lecture 14: Sensitivity Analysis

25 Oct. 08:00

6 Sensitivity Analysis

As usual, we start with the primal and the dual

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

with an optimal basic partition β, η such that

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

As previously seen. The dual feasibility is

$$\overline{c}_\eta := c_\eta - c_\beta^T A_\beta^{-1} A_\eta = c_\eta - \overline{y}^T A_\eta \geq \vec{0}.$$

6.1 Local Analysis

6.1.1 Change b on the right-hand side

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.

6.1.2 Objective Value

Now, the objective value is

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

6.1.3 Analysis

Let f be

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

where

$$f : \mathbb{R}^m \to \mathbb{R}.$$

We see that since the optimal objective value is equal for the dual of P_b, then $f(b) = y^T b$. 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

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

Reality Check. We see that

$$L_i \leq 0 \leq U_i.$$

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 $U_i \coloneqq \infty$.

6.2 Global Analysis

We start with a theorem.

Theorem 6.1. The domain of f is a convex set.

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

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

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

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

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

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

∎

We now introduce the convexity of a function.

Definition 6.1. We say that f is a *convex function* on a convex domain S if

$$x^1, x^2 \in S \text{ and } 0 < \lambda < 1,$$

then

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

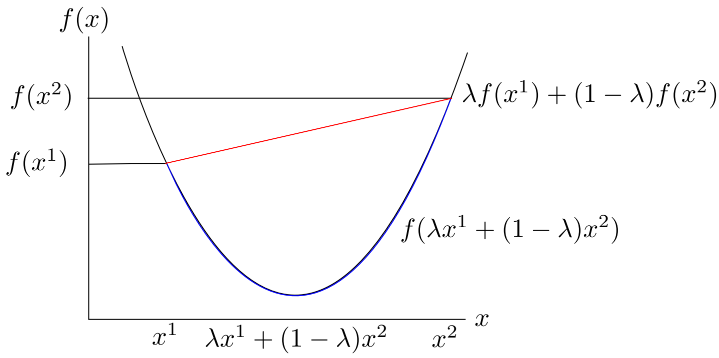


Figure 14: Convex Function

6.2.1 Affine Function

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

Definition 6.2. A function $f\colon \mathbb{R}^m \to \mathbb{R}$ is *affine* if

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

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

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

Definition 6.3. A function $f\colon \mathbb{R}^m \to \mathbb{R}$ is a *convex piece-wise linear function* if f **is the point-wise maximum of affine functions.**

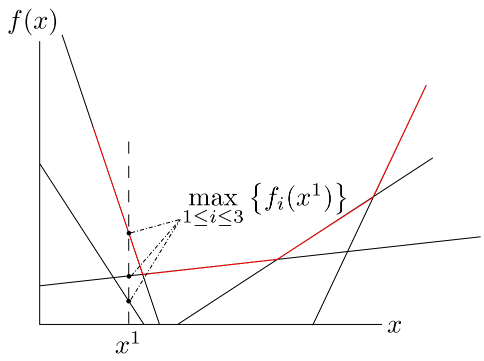


Figure 15: Convex Piece-wise Linear Function

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

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

Theorem 6.2. The point-wise maximum of affine functions is a convex function.

Proof. We see that

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

■

Remark. The second equality holds since

$$\begin{aligned}
&\max_{1\le i\le K} \left\{a_{i0} + \sum_{l=1}^{m} a_{il}(\lambda u_l^1 + (1-\lambda)u_l^2)\right\} \\
&= \max_{1\le i\le K} \left\{\lambda a_{i0} + (1-\lambda)a_{i0} + \sum_{l=1}^{m} a_{il}(\lambda u_l^1 + (1-\lambda)u_l^2)\right\}
\end{aligned}$$

Lecture 15: Sensitivity Analysis

27 Oct. 08:00

6.3 More on Local Analysis

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

$$\overline{c}_\eta \coloneqq \boldsymbol{c_\eta^T} - \boldsymbol{c_\beta^T} A_\beta^{-1}\boldsymbol{A_\eta} \geq \vec{0},$$

we see that $\boldsymbol{c}, \boldsymbol{b}, \boldsymbol{A_\eta}$ 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.

6.3.1 Change A_η on the right-hand side

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

$$\begin{aligned} \overline{c}'_{\eta_j} &= c_{\eta_j} - \underline{c_\beta^T A_\beta^{-1}} \left(A_{\eta_j} + \Delta e_i\right) \\ &= c_{\eta_j} - \overline{y}^T(A_{\eta_j} + \Delta e_i) \\ &= \overline{c}_{\eta_j} - \Delta\overline{y}_i \underset{\text{want}}{\geq} 0. \end{aligned}$$

Hence, the condition becomes

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

6.3.2 Change c on the right-hand side

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

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

Hence, the condition becomes

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

Now, for c_{β_i},

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

Then

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

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

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

Separate them, we see

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

Equivalently,

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

and

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

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

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

6.3.3 Change b on the right-hand side, but in two entries

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

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

Then

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

Writing things separately, we have

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

where $H := A_\beta^{-1}$. 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 variables 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

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

Originally, the objective value is

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

after considering the changes, we have

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

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

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

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

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

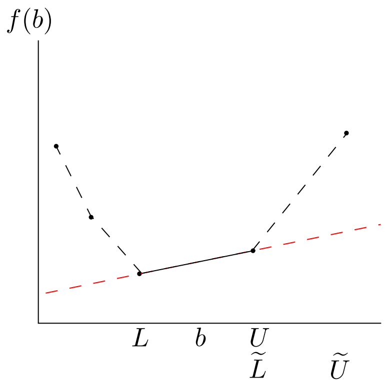


Figure 16: Local Analysis

6.4 More on Global Analysis

Still, consider the primal and dual pair

$$\begin{aligned} f(b) = \min \ & c^T x & \qquad \max \ & y^T b \\ & Ax = b & & y^T A \leq c^T. \\ (P_b) \quad & x \geq 0 & (D_b) & \end{aligned}$$

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

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

Consider

$$\begin{aligned} g(c) = \min \quad & c^T x \\ & Ax = b \\ (P_c) \quad & x \geq 0 \end{aligned}$$

where g is a piece-wise linear concave function in c.

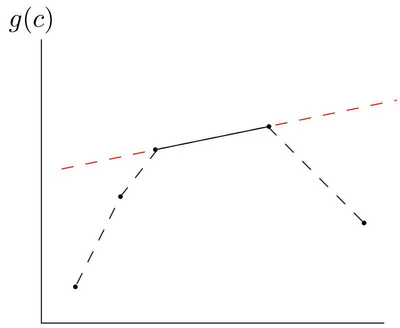


Figure 17: The dual version

We see that D_b is equivalence to

$$\begin{aligned} -\min \quad & -(y^+ - y^-)^T b \\ & (y^+ - y^-)^T A + IS^T = c^T \\ & y^+ \geq 0, \; y^- \geq 0 \end{aligned}$$

Lecture 16: Large-Scale Linear Optimization 1 Nov. 08:00

7 Large-Scale Linear Optimization

Let's first look at an example.

Example. Consider a constraint matrix

$$\begin{pmatrix} \multicolumn{5}{c}{[\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad]} \\ \begin{bmatrix} \quad \\ \quad \end{bmatrix} & & & & \\ & [\quad] & & 0 & \\ & & \begin{bmatrix} \quad \\ \quad \\ \quad \end{bmatrix} & & \\ & 0 & & \ddots & \\ & & & & \begin{bmatrix} \quad \\ \quad \end{bmatrix} \end{pmatrix}$$

We see that if the first constraint(the first row) doesn't exist, then the problem decomposes to those small block matrix corresponds to some smaller, easier linear optimization problems, and we can solve it very quickly.

Note. We called the above matrix as **Nearly Separates**.

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

7.1 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.

7.1.1 Representation Theorem

Let P be

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

Theorem 7.1 (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

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

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

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

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

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

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

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

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

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

$$\begin{aligned} w^T\hat{x}+t &> 0 & & -w^T\hat{x} &< t \\ w^T\hat{x}^j+t &\le 0 & \iff & -w^T\hat{x}^j &\ge \hat{t} \text{ for } j\in\mathcal{J} \\ w^T\hat{z}^k &\le \vec{0} & & -w^T\hat{z}^K &\ge 0 \text{ for } k\in\mathcal{K}. \end{aligned}$$

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

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

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 $-\hat{w}^T\hat{x}\le\hat{t}$ and $-\hat{w}^T\hat{x}^j\ge\hat{t}$, 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 $-\hat{w}\hat{z}^k\ge 0$, we see that this will never happen, hence the theorem is proved.

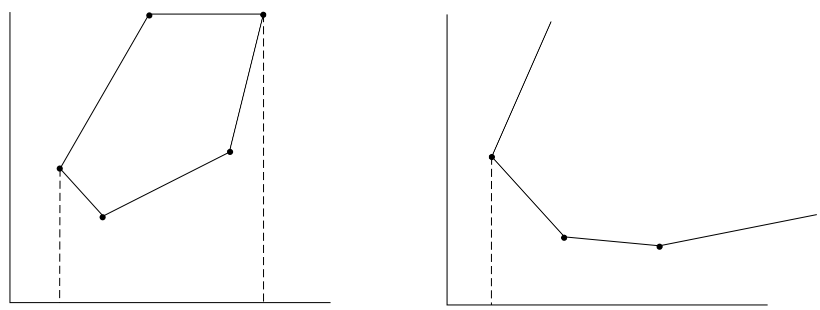


Figure 18: bounded and unbounded case in Simplex Algorithm

■

With this representation theorem, consider

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

Then by

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

we turn the linear problem into

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

Furthermore, this is equivalent to

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

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

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

Theorem 7.2 (The decomposition theorem). Let

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

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

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

Remark. We think of E being a *complicated* constraints 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. Indeed, we will soon see that we can turn M into a standard form problem without increasing complexity.

7.2 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 $\overline{M}$, 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

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

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

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

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

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

Answer. Consider

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

Lecture 17: Large-Scale Linear Optimization

3 Nov. 08:00

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

1. What's the reduced coast of s_i?

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

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

2. What's the reduced cost of λ_j?

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

   We consider a sub problem

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

   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?

3. What's the reduced cost of μ^k?

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

Again, consider a sub problem

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

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 $\implies$ λ_j to enter the basis.

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

iii. unbounded $\implies$ we get a $\overline{z}$ that is a basic ray with $c^T\overline{z} < 0$, 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 $\overline{M}$ will consists 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

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

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

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

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

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

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

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

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 $B^{-1}\begin{pmatrix} h \\ 1 \end{pmatrix} \geq \vec{0} \implies$ 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 nonbasic variable with negative reduced cost?
$$\overline{c}_j \coloneqq c_j - \overline{y}^T A_{\eta_j} < 0.$$
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**.

As previously seen. The Decomposition Algorithm. We change the step 0 and 2 of the original Simplex Algorithm into the following.

0. Reformulate Q as M and apply The Simplex Algorithm to M, where

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

and

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

2. Solve the sub-problem

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

- Optimal & Obj.val $< 0 \implies$ a λ variable can enter the basis.
- Optimal & Obj.val $> 0 \implies$ have an optimal for M.
- Unbounded $\implies$ 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.

7.3 Lagrangian Relaxation

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

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

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?

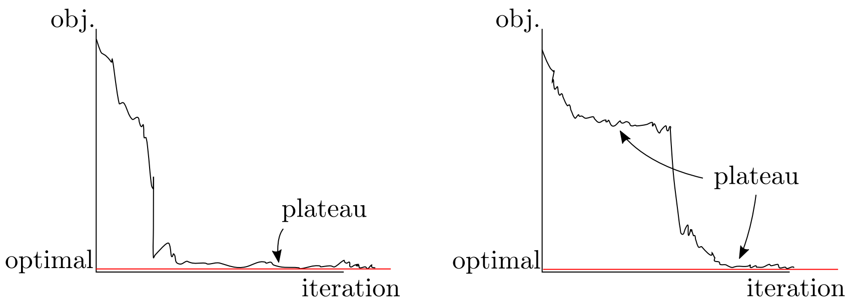


Figure 19: Early Arrival, can we?

7.3.1 Laguargian Bounds

We choose $\hat{y} \geq \vec{0}$, such that

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

where L stands for Lagrange.

Intuition. We see that 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 7.1. 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

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

∎

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

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

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

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

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

Theorem 7.3. 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.

Note. In above, we want to find

$$\begin{aligned} z = \max \ & v(y) \\ & y \geq 0. \end{aligned}$$

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

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

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

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

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

From Strong Duality Theorem(Theorem 5.3) for Q,

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

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

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

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

$$\hat{\pi}^T \underbrace{A x^*}_{b} \leq (c^T - \hat{y}^T E)x^* \iff \hat{\pi}^T b \leq (c^T - \hat{y}^T E)x^*.$$

We conclude

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

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

$$\hat{y}^T h + \min\ (c^T - \hat{y}^T E)x,$$

with $(c^T - \hat{y}^T E)x^* \geq \hat{\pi}^T b$, we see that the objective value of x^* in $L_{\hat{y}}$ is equal to $\hat{y}h + \hat{\pi}^T b$, which implies that x^* is optimal in $L_{\hat{y}}$ from Weak Duality Theorem(Theorem 1.1).

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

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

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

$$\begin{aligned} \max\ & v(y) \\ & y \geq 0 \end{aligned}$$

and $v(\hat{y}) = z$. ■

Theorem 7.4. 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. We see that compare to the last theorem, we now try to say something *backwards*. But we immediately see that it suffers from the situations depicts as follows.

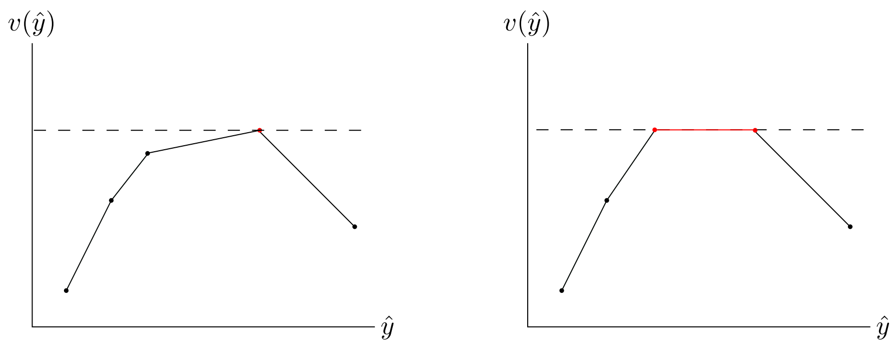


Figure 20: There may exist several $\hat{y}$!

Lecture 19: Lagrangian Relaxation

10 Nov. 08:00

As previously seen.

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

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

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

which we called it a **Lagrangian subproblem**.

Note. We see that for $\hat{y} \geq \vec{0}$, $v(\hat{y}) \leq z$. Now, the goal is to solve the **Lagrangian dual**, which is

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

to get a lower bound for the original problem.(Notice that this is the maximum of the dual!)

Now, we try to proof Theorem 7.4, which is the *partial* converse of Theorem 7.3.

Proof. Recall that

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

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

7.3.2 Solving the Lagrangian Dual

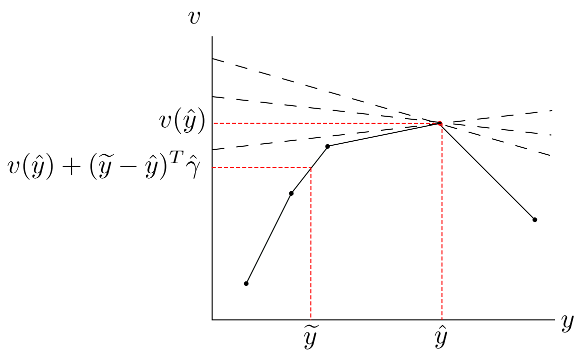
The theorem we just saw provides a simple way to calculate a lower bound on z by solving a potentially easier linear optimization problem. But we see that the bound depends on the choice of $\hat{y} \geq 0$. This push us to find the best such $\hat{y}$, and we indeed can solve this by solving the so-called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

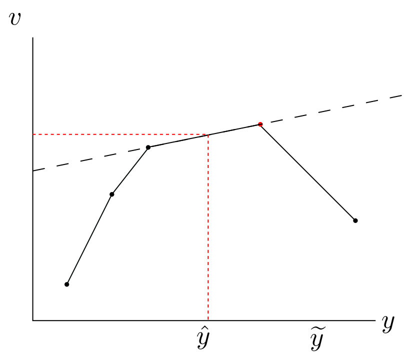
One may want to use some calculus technique to solve for such maximizing problem, but since v is not a smooth function, rather a piece-wise linear function, hence we need to introduce the concept of *subgradient*. Before we formally introduce it, we first see a theorem.

Theorem 7.5. 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

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

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 so-called the *subgradient*. Given $\widetilde{y}$ and $\hat{y}$, we choose $\hat{\gamma}$ such that the linear estimation $v(\hat{y})+(\widetilde{y}-\hat{y})^T\hat{\gamma}$ 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) v at $\hat{y}$.

Mathematically, we have

Definition 7.1. For a concave function f such that

$$f\colon I \to \mathbb{R},$$

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 subgradient theorem, we can then develop an algorithm to utilize this.

7.3.3 Projected Subgradient Optimization Algorithm

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

0. Start with any $\mathbb{R}^m \ni \hat{y}^T \geq \vec{0}$. Let $k := 1$
1. Solve $L_{\hat{y}^k}$ to get $\hat{x}^k$
2. Calculate the subgradient $\hat{\gamma}^k$ by

$$\hat{\gamma}^k := h - E\hat{x}^k.$$

3. Let $\hat{y}^{k+1} \leftarrow \mathrm{Proj}_{\mathbb{R}^m_+}(\hat{y}^k + \lambda_k\hat{\gamma}^k)$
4. Let $k \leftarrow k+1$ & **GOTO 1.**

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 step 3. 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

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

    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

17 Nov. 08:00

As previously seen. We have already shown the algorithm of projected subgradient optimization, 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.

Lemma 7.2. 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 $w^{k+1} \coloneqq \hat{y}^k + \lambda_k \hat{\gamma}^k$. Then for $k \geq 1$,

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

where the first inequality follows from the triangle inequality, and the last inequality follows from the definition of subgradient. From the argument above, we just do some *telescoping* and see the lemma holds. ■

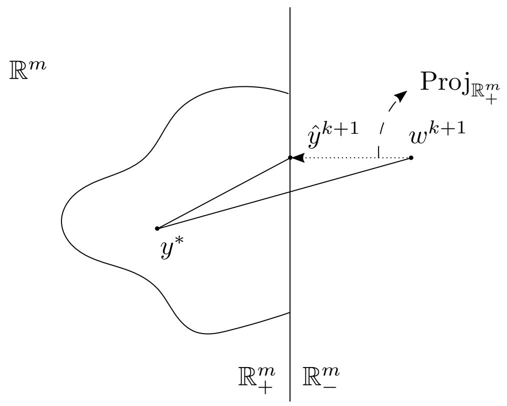


Figure 21: Triangle Inequality for $\mathrm{Proj}_{\mathbb{R}^m_+} w^{k+1} = \hat{y}^{k+1}$.

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

Theorem 7.6. 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 previous lemma, 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

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

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

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

With Γ, the inequality becomes

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

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 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 $v(y^*) - v_k^*$.

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

Consider

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

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

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

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

22 Nov. 08:00

7.4 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.

Problem. Contrarily, *Cutting-Stock Problem* can be nicely approximated by **rounding down**. Consider we have rolls of paper of width W, with the demand widths being $w_1, w_2, \ldots, w_m < W$ 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 the Cutting-Stock problem is simply

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

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 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 CSP exactly. Denotes the dual variables of 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:

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

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

- x_j:

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

  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,

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

This is just a *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 $\Theta(Wm)$.

Note. Note that we assume W and w_i are integer.

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.

Lecture 22: Optimization of Integer Variables

29 Nov. 08:00

8 Integer-Linear Optimization

Let first see some common pitfalls of integer programming.

- 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.

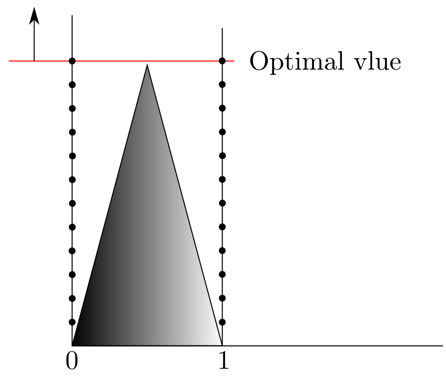


Figure 22: Pitfall of Integer Programming

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.

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

  Then, there are two feasible solutions one can observe immediately, namely

$$x_1 = x_2 = \ldots = x_n = \frac{1}{2};$$

  and

$$x_1 = 1, x_2 = \ldots = x_n = 0.$$

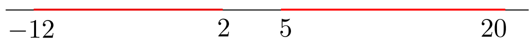
  It's then really hard to tell which is better. But again, if the right-hand side is 2 for the first constraint, then the problem is gone.

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.

8.1 Modeling Techniques

Example. Consider the following constraints.

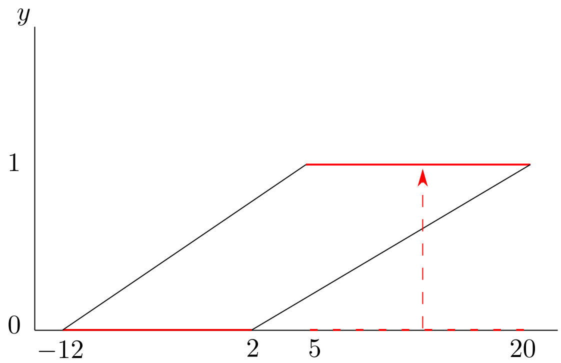
$$-12 \leq x \leq 2 \ \vee \ 5 \leq x \leq 20.$$



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

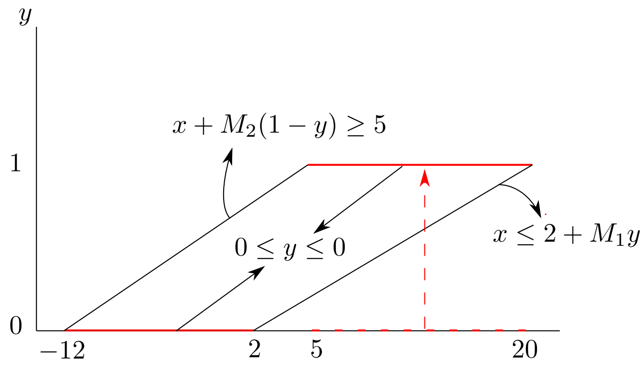
$$\begin{aligned} -12 \leq x &\leq 20 \\ 0 \leq y &\leq 1, \text{ integer} \\ x &\leq 2 + M_1 y \\ x + M_2(1-y) &\geq 5, \end{aligned}$$

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 $M_1 \coloneqq 18$, then

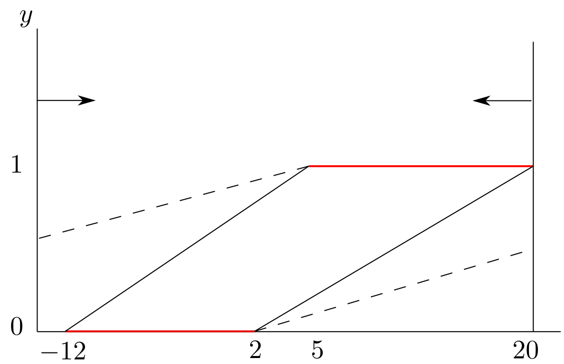
$$\begin{cases} y = 0, & x \leq 2 \\ y = 1, & x \leq 20. \end{cases}$$

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 $M_2 \coloneqq 17$.

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.

8.2 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

$$x_{ij} := \text{proportion of customer } j \text{ demand satisfied facility } i,$$

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

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} f_i y_i + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} & \\ & \sum_{i=1}^{m} x_{ij} = 1, && \text{for } j = 1, \ldots, n \\ & -y_i + x_{ij} \leq 0, && \text{for } i = 1, \ldots, m,\ j = 1, \ldots, n \\ & 0 \leq y_i \leq 1 \text{ integers}, && \text{for } i = 1, \ldots, m \\ & x_{ij} \geq 0, && \text{for } i = 1, \ldots, m,\ j = 1, \ldots, n. \end{aligned}$$

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

$$x_{ij} > 0 \implies y_i = 1$$

for any j. It is equivalent to say

$$\sum_{j=1}^{n} x_{ij} > 0 \implies y_i = 1.$$

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

$$\sum_{j=1}^{n} (x_{ij} \leq y_i) \iff \sum_{j=1}^{n} x_{ij} \leq n y_i,$$

we see that the weak constraint is just the sum over all strong constraint. In other words, we have

$$x_{ij} \leq y_i \implies \sum_{j=1}^{n} x_{ij} \leq n y_i.$$

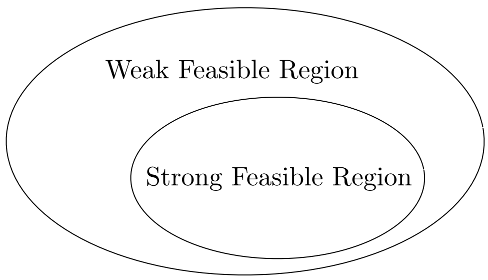
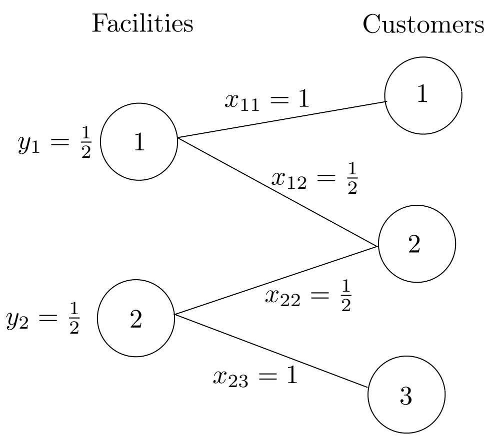


Figure 23: Venn diagram of Strong/Weak Feasible Region

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 $x_{11} \nleq y_1$, but

$$x_{11} + x_{12} \leq 3y_1$$

and

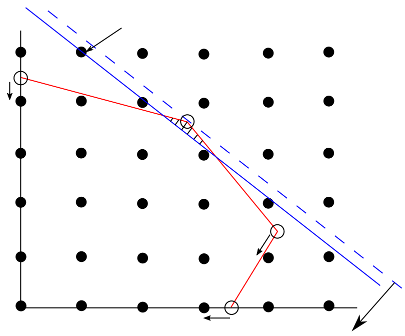
$$x_{22} + x_{23} \leq 3y_2.$$

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!

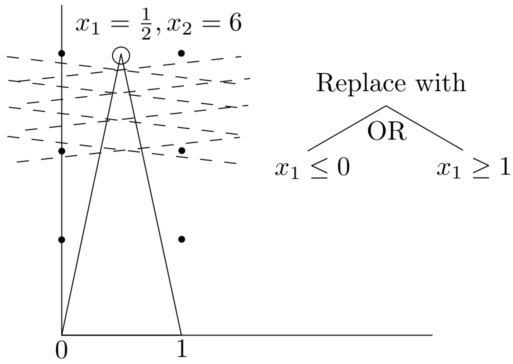
It's worth noting that the process of un-summing from the weak constraint to the strong constraint is called *disaggregation*.

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 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 method, and then on top of that, solve the remaining problem by cutting-plane algorithm.

Lecture 23: Branch and Bound

01 Dec. 08:00

8.3 Branch and Bound

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,

$$\begin{array}{rl} \max & y^T b \\ & y^T A \leq c^T \\ (D_{\mathcal{I}}) & y \in \mathbb{R}^m (y_i \in \mathbb{Z} \text{ for } i \in \mathcal{I}), \end{array}$$

where $\mathcal{I} \subseteq \{1, 2, \ldots, m\}$. By taking the dual, we have

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

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 $D_{\mathcal{I}}$.
- LB: The current best lower bound on z such that $\text{LB} \leq z$.
- $\overline{y}_{\text{LB}}$: 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

$$\text{LB} \coloneqq -\infty$$

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 subproblem on $\mathcal{L}$.

Initially, we have

$$\mathcal{L} \coloneqq \{D_{\mathcal{I}}\}.$$

And we stop if

$$\mathcal{L} = \varnothing,$$

since this implies $z = \text{LB}$.

The general procedure is to take some problem $\widetilde{D}_{\mathcal{I}}$ from $\mathcal{L}$ and remove it, and then solve its linear programming $\widetilde{D}$. Then we see

- If $\widetilde{D}$ is infeasible, then do nothing.
- Otherwise, let $\overline{y}$ be its basic optimal solution.
  - If $\overline{y}^T b \leq \text{LB}$, then do nothing.
  - Otherwise,
    - If $\overline{y}_i^T \in \mathbb{Z}$ for $i \in \mathcal{I}$, then $\overline{y}$ solves $\widetilde{D}_{\mathcal{I}}$. Let

      $$\text{LB} \coloneqq \overline{y}^T b \text{ and } \overline{y}_{\text{LB}} \coloneqq \overline{y}.$$

    - If $\overline{y}_i^T \notin \mathbb{Z}$ for some $i \in \mathcal{I}$,
      - Select $i \in \mathcal{I}$ such that $\overline{y}_i \notin \mathbb{Z}$.
      - Place two *child* problems on L:

      1. *Down branch*: $\widetilde{D}_{\mathcal{I}}$ with

         $$y_i \leq \lfloor \overline{y}_i \rfloor .$$

2. *Up branch*: $\widetilde{D}_{\mathcal{I}}$ with

$$y_i \geq \lceil \overline{y}_i \rceil .$$

Note. To match the form, we use $-y_i \leq -\lceil \overline{y}_i \rceil$.

Process: $\widetilde{D}_{\mathcal{I}}$(problem chosen from $\mathcal{L}$)

- Solve linear programming relaxation $\widetilde{D}$

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

What we're really doing is solving P by simplex algorithm. Let $\overline{y}^T := c_\beta^T A_\beta^T$.

– *Down branch*:

$$\begin{array}{llll} & \max\ y^T b & & \min\ c^T x + \lfloor \overline{y}_i \rfloor x_{down} \\ & \quad y^T A \leq c^T & & \quad Ax + e_i x_{down} = b \\ (D) & \quad y_i \leq \lfloor \overline{y}_i \rfloor & (P) & \quad x \geq 0, x_{down} \geq 0. \end{array}$$

The reduced cost of x_{down} is

$$\lfloor \overline{y}_i \rfloor - \overline{y}^T e_i = \lfloor \overline{y}_i \rfloor - \overline{y}_i.$$

If this is negative, then x_{down} enters the basis, which happens when $\overline{y}_i \notin \mathbb{Z}$.

– *Up branch*:

$$\begin{aligned} \min\ & c^T x - \lceil \overline{y}_i \rceil x_{up} \\ & Ax - e_i x_{up} = b \\ & x \geq 0, x_{up} \geq 0. \end{aligned}$$

The reduced cost of x_{up} is

$$-\lceil \overline{y}_i \rceil - \overline{y}^T(-e_i) = \overline{y}_i - \lceil \overline{y}_i \rceil .$$

If this is negative, then x_{up} enters the basis.

Remark. In practice,

– When we solve a child, our best wish is that optimal value $\leq$ LB.(then we don't have to branch)

– As we solve the child, we can stop once its objective value is at or below LB.

8.3.1 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.

8.3.2 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) $\cong$ BFS(Breadth First Search)

   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) $\cong$ DFS(Depth First Search)

   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.

8.3.3 Choosing Branching Variable

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

06 Dec. 08:00

8.4 Cutting Planes

As previously seen. We focus on the problem in the form of

$$\begin{aligned} & \max \ y^T b \\ & \quad y^T A \leq c^T \\ (D_{\mathfrak{X}}) \quad & y_i \text{ integer for } i = 1, \dots, m. \end{aligned}$$

Note. Compare to what we have seen, now we require all y_i be integer. Further, as before, we also let P be

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

Remark. We assume that the data is all integer.

Now, we choose $w \in \mathbb{R}^n$, $w \geq 0$. Then the constraint becomes

$$y^T (Aw) \leq c^T w.$$

Remark. This valid for all y such that

$$y^T A \leq c^T,$$

no matter it's integer or not.

Suppose $Aw \in \mathbb{Z}^m$. With the fact that $y \in \mathbb{Z}^m$, then for

$$y^T (Aw) \leq c^T w,$$

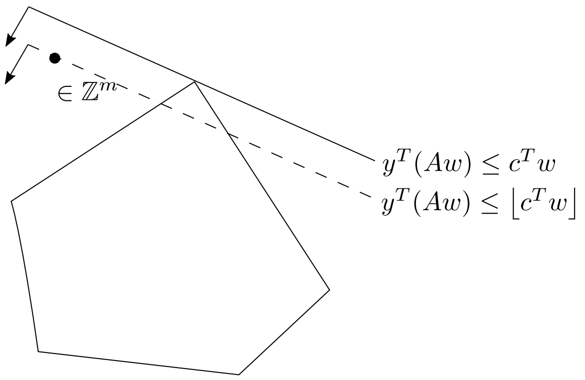
we can actually get

$$y^T (Aw) \leq \left\lfloor c^T w \right\rfloor.$$

Remark. This is valid for all y that satisfies

$$y^T (Aw) \leq c^T w$$

and $y \in \mathbb{Z}^m$.



We now solve P and get an optimal basis β. Consider

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

Notice that if $\overline{y} \in \mathbb{Z}^m$, then $\overline{y}$ solves $D_{\mathfrak{X}}$. Otherwise, suppose $\overline{y}_i \notin \mathbb{Z}$, then let

$$\widetilde{b} := e_i + A_\beta r \in \mathbb{Z}^m,$$

where $r \in \mathbb{Z}^m$. We then see a theorem.

Theorem 8.1. If $\overline{y}^T\widetilde{b} \notin \mathbb{Z}$, then

$$y^T\widetilde{b} \leq \left\lfloor \overline{y}^T\widetilde{b} \right\rfloor$$

cuts off $\overline{y}$.

Proof. Since

$$\overline{y}^T\widetilde{b} = \overline{y}^T(e_i + A_\beta r) = \overline{y}_i^T + \overline{y}^T A_\beta r = \overline{y}_i^T + c_\beta^T A_\beta^{-1} A_\beta r = \overline{y}_i^T + c_\beta^T r$$

We see that $\overline{y}_i \notin \mathbb{Z}$, $c_\beta^T, r \in \mathbb{Z}$, hence we have

$$\overline{y}^T\widetilde{b} = \overline{y}_i + c_\beta^T r \notin \mathbb{Z}.$$

Now, we need to check that $y^T\widetilde{b} \geq \left\lfloor \overline{y}^T\widetilde{b} \right\rfloor$ is satisfied by $\overline{y}$.

Intuition. Consider if the inequality is

$$\vec{0}^T y \leq -1,$$

then it makes no sense.

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 $K = 1, \ldots, m$.

Instead of considering $y^T A \leq c^T$, we consider $y^T A_\beta \leq c_\beta^T$. Then we have

$$\left(y^T A_\beta\right)\left(H_{\cdot i} + r\right) \leq c_\beta^T \left(H_{\cdot i} + r\right).$$

This is equivalence to

$$\left(y^T A_\beta\right)\left(A_\beta^{-1} e_i + r\right) \leq c_\beta^T \left(A_\beta^{-1} e_i + r\right).$$

After expanding, we have

$$y_i + y^T A_\beta r \leq \overline{y}_i + c_\beta^T r,$$

which can be written as

$$y^T \left(e_i + A_\beta r\right) \leq \overline{y}^T \left(e_i + A_\beta r\right)$$

since $\overline{y}^T = c_\beta^T A_\beta^{-1}$. Then we see

$$y^T \widetilde{b} \leq \left\lfloor \overline{y}^T \widetilde{b} \right\rfloor.$$

Lastly, we need Aw are all integers. This is true since

$$A_\beta w = A_\beta \left(A_\beta^{-1} e_i + r\right) = e_i + A_\beta r \in \mathbb{Z}^m.$$

■

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.

$$\begin{aligned} \min\ & y_{n+1} \\ & 2y_1 + 2y_2 + \ldots + 2y_n + y_{n+1} = \underbrace{n}_{\text{odd}} \\ & 0 \leq y_i \leq 1 \text{ for } i = 1, \ldots, n+1, \text{ integer.} \end{aligned}$$

We see that the optimum has $y_{n+1} = 1$.

If $n = 17$. Then we can let

$$y_{18} = 0, \quad y_1 = y_2 = \ldots = y_8 = 1, \quad y_9 = \frac{1}{2}, \quad y_{10} = \ldots = y_{17} = 0.$$

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.

Appendix

This note is completed in LaTeXwith Inkscape, in case of anyone is interested, please check out this blog[1] together with my configuration[2].

[1] https://castel.dev/
[2] https://github.com/sleepymalc/VSCode-LaTeX-Inkscape

References

[Lee22] Jon Lee. *A First Course in Linear Optimization"*. Reex Press, 2022. URL: https://github.com/jon77lee/JLee_LinearOptimizationBook.