Note on Optimization

Zepeng CHEN

June 21, 2022

Contents

Chapter 1 Linear Programming

1.1 What is Linear Programming

1.1.1 Standard LP

Definition 1.1 (Standard LP)

$$\begin{aligned} &\min c^T x \\ &s.t. Ax \leq b, x \geq 0 \end{aligned} \tag{1.1}$$

Note on Perspective of geometry *c determines the direction of fastest increase, cx denote the hyperplane with the direction c of fastest increase, and A define the feasible region by clarifying the intersection of half spaces.*

Note on Example of standard LP transformation

1. *"Max" to "Min": Adding negative sign to the objective.*
2. $a_{i1}x_1 + \cdots + a_{in}x_n \geqslant b_i \Rightarrow a_{i1}x_1 + \cdots + a_{in}x_n - x_{n+1} = b_i$*, where x_{n+1} is the surplus variable.*
3. $a_{i1}x_1 + \cdots + a_{in}x_n \leqslant b_i \Rightarrow a_{i1}x_1 + \cdots + a_{in}x_n + x_{n+1} = b_i$*, where x_{n+1} is the slack variable.*
4. *Free variable:* $x_i = x_i^+ - x_i^-, x_i^+ \geqslant 0, x_i^- \geqslant 0$
5. *Absolute variable:* $|x_j| = x_j^+ + x_j^-, x_j = x_j^+ - x_j^-$*, adding a constraint* $x_j^+ \cdot x_j^- = 0$*, note that this constraint can be neglected if* $c_j \geq 0$
6. a^+ *or* $\max(a, 0)$*:* $\max 170(3x-240)^+ - 238(240-3x)^+$ *to the following LP. If* $3x-240 > 0$*,* y_1 *increases to* $3x-240$ *and* y_2 *decreases to 0 (*$y_1 > y_2$*), If* $3x - 240 < 0$*,* y_1 *increases to 0,* y_2 *decreases to* $240 - 3x$ *(*$y_1 < y_2$*).*

$$\begin{aligned} &\max 170 y_1 - 238 y_2 \\ &s.t. 3x - 240 = y_1 - y_2, y_1, y_2 \geq 0 \end{aligned} \tag{1.2}$$

7. *Quantity discount: For example, if* $p = 4000$ *for* $x < 30$*,* $p = 2000$ *for* $30 \leq x < 50$ *and* $p = 1500$ *for* $x > 50$*. Using two binary variables:* σ_2, σ_3*. If* $\sigma_2, \sigma_3 = 0, 0$*, it means that* $x_2 = x_3 = 0$*. And* $\sigma_2, \sigma_3 = 0, 1$ *does not exist. If* $\sigma_2, \sigma_3 = 1, 0$*, it means that*

$0 \le x \le 20, x_3 = 0$. *If* $\sigma_2, \sigma_3 = 1, 1$, *it means that* $0 \le x \le 20, x_3 \le M$.

$$
\begin{aligned}
&\min 4000x_1 + 2000x_2 + 1500x_3 \\
&s.t. x_1 + x_2 + x_3 = Demand \\
&x_2 \le 20\sigma_2 \\
&x_1 \ge 30\sigma_2 \\
&x_2 \ge 20\sigma_3 \\
&x_3 \le M\sigma_3
\end{aligned}
\tag{1.3}
$$

8. $b_i < 0$: *Adding negative sign to the whole constraints.*
9. $x \le 0$: *Let* $x' = -x$
10. $l \le x \le u$

$$
\begin{aligned}
\min c^T x \qquad & \min c^T x^+ - c^T x^- \\
s.t. Ax \le b \qquad & s.t. Ax^+ - Ax^- + s_1 = b \\
l \le x \le u \qquad & x^+ - x^- + s_2 = u \\
& x^+ - x^- - s_3 = l \\
& x^+, x^-, s_1, s_2, s_3 \ge 0
\end{aligned}
\tag{1.4}
$$

11. *Linear Fractional Programming: Define* $y = \frac{x}{e^T x + f}$ *and* $z = \frac{1}{e^T x + f}$, *here* z *in LP1 cannot be zero, though* z *in LP2 can be zero, we can show that these two are equivalent. (1)* z^* *in LP2 is not zero, then the optimal solution are the same; (2)* z^* *in LP2 is zero, then it means* $e^T x^* + f \to \infty$.

$$
\begin{array}{llll}
& \textbf{\textit{LP1}} & & \textbf{\textit{LP2}} \\
\max\limits_{x} & \dfrac{c^T x + d}{e^T x + f} & \min\limits_{y,z} & c^T y + dz \\
s.t. & Ax \le b & s.t. & Ay - bz \le 0 \\
& e^T x + f > 0 & & e^T y + fz = 1 \\
& & & z \ge 0
\end{array}
\tag{1.5}
$$

1.1.2 Basic and Optimal Solution

Definition 1.2 (Feasible, Basic, Optimal, Degenerate Solution)

1. *Feasible solution:= a solution which satisfies the constraints* $Ax = b, x \ge 0$.
2. *Basic solution:=* $x = (x_B, x_N)$, *where* x_B *is linear independent* $m \times m$ *matrix,* x_N *is* $m \times n - m$ *matrix, the solution attained by set* x_N *to zero.*
3. *Basic feasible solution:= A solution which is feasible and basic.*
4. *Degenerate basic solution:= A basic solution with one or more basic variables has the value zero.*
5. *Optimal feasible solution:= A feasible solution that achieves the minimum value.*

6. *Optimal basic feasible solution:= A optimal feasible solution which is also basic.*

Note on Geometric Interpretation of Degenerate Solution *In the two-dimensional space, degenerate solution denotes the intersection of three or more lines. In the three-dimensional space, degenerate solution denotes the intersection of four or more planes. The nature of degenerate solution is that it remains the same point after pivoting.*

Note on *Several optimal solutions does not mean there exist at least two basic feasible solution that are optimal, e.g.* $\{(x, y) \in \mathbb{R}^2 \mid -x + y = 0, \quad x, y \geq 0\}$. *Only one basic feasible solution, but the whole line is optimal.*

1.1.3 LP with Bounded Variables

$$
\begin{aligned}
\min c^T x \qquad & \Rightarrow c^T(x_B, x_N) \\
\text{s.t.} Ax = b \qquad & \Rightarrow I x_B + \bar{A} x_N = \bar{b} \\
l \leq x \leq u &
\end{aligned}
\tag{1.6}
$$

Definition 1.3 (Generalized definition and condition)

- *Basic solution* $:= x_N$ *equal to either the lower bound or upper bound.*
- *Degenerate basic solution* $:=$ *one or more* $x_B = l$ *or* u.
- *Optimality condition* $:=$ *A basic solution* $x = (x_B^*, x_N^*)$ *is optimal if*
  - $l_B \leq x_B^* \leq u_B$. *(feasibility)*
  - $r_j \geq 0 \quad \forall j \in L = \{j \in N | x_j^* = l_j\}$ *and* $r_j \leq 0 \quad \forall j \in U = \{j \in N | x_j^* = u_j\}$.

1.2 Feasibility

Definition 1.4 (Feasible Direction)

Let x *be an element of a polyhedron* P. *A vector* $d \in R^n$ *is said to be a feasible direction at* x, *if there exists a positive scalar* θ *for which* $x + \theta d \in P$.

Lemma 1.1 (Feasible Direction (Bertsimas et al., 1997, P. 129))

For polyhedron $P = \{x \in \Re^n \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, *a vector* $d \in R^n$ *is a feasible direction at* x *iff* $Ad = 0$ *and* $d_i \geq 0$ *for every* i *such that* $x_i = 0$.

1.3 Optimality and Uniqueness

Proposition 1.1 (Interior point and Optimality)

$$\min c^T x$$
$$s.t. Ax \leq b, x \geq 0 \tag{1.7}$$

The point satisfies that $Ax_0 < b, x_0 > 0$ cannot be an optimal solution.

Proof Suppose for the sake of contradiction that there exists another point $x_0 \pm \varepsilon c > 0$ and $A(x_0 \pm \varepsilon c) < b$, then we show that $c^T(x_0 \pm \varepsilon c) = c^T x_0 \pm \varepsilon ||c||$, that is, the new point is more optimal than the former one.

Next we construct $\varepsilon > 0$ that we want, for any $x_0 + \varepsilon c > 0$,x_0, $x_{0i} > 0$ must hold. However, $c_i < 0$ may occur, we let $\varepsilon < \min\{\frac{x_i}{|c_i|}\} \forall c_i < 0$. For $A(x_0 + \varepsilon c) < b$, in each row, we want $\sum_{i=1}^n a_i(x_i + \varepsilon c_i) < b_i$. Thus, we can let $\varepsilon < \min\{\frac{b_i - \sum_{i=1}^n a_i x_i}{\sum_{i=1}^n a_i c_i}\}$. ∎

Theorem 1.1 (Optimality Conditions (Bertsimas et al., 1997, P. 129))

Consider the problem of minimizing $c^T x$ over a polyhedron P

1. *A feasible solution x is optimal iff $c^T d \geq 0$ for every feasible direction d at x*
2. *A feasible solution x is unique optimal iff $c^T d > 0$ for every nonzero feasible direction d at x*

Theorem 1.2 (Conditions for a unique optimum (Bertsimas et al., 1997, P. 129))

Let X be a basic feasible solution with basis B

1. *If the reduced cost of every nonbasic variable is positive, then x is the unique optimal solution.*
2. *If x is the unique optimal solution and is nondegenerate, then the reduced cost of every nonbasic variable is positive.*

1.4 LP based on Basis and Topological Space

1.4.1 Caratheodory's theorem

Proposition 1.2 (Caratheodory's theorem)

Let $\mathbf{A}_1, \ldots, \mathbf{A}_n$ be a collection of vectors in R^m, Let

$$C = \left\{ \sum_{i=1}^n \lambda_i \mathbf{A}_i \mid \lambda_1, \ldots, \lambda_n \geq 0 \right\}$$

Then any element of C can be expressed in the form $\sum_{i=1}^n \lambda_i \mathbf{A}_i$, with $\lambda_i \geq 0$, and with at most m of the coefficients λ_i being nonzero.

Proof When $n \leq m$, obviously the condition holds. When $n > m$, consider a polyhedron

$$\Lambda = \left\{ (\lambda_1, \ldots, \lambda_n) \in \Re^n \mid \sum_{i=1}^{n} \lambda_i \mathbf{A}_i = \mathbf{y}, \lambda_1, \ldots, \lambda_n \geq 0 \right\}$$

This is a standard LP, thus there is at least a extreme point, that is, a basic feasible solution $\lambda^* = (\lambda_1^*, \ldots, \lambda_n^*)$. Note that we have at most m linear independent vectors among A_i, thus a basic feasible solution has at least $n - m$ zero components, which means there are at most m non-zero components in λ^*. ■

1.4.2 Feasibility to Basic Feasibility

Theorem 1.3 (Fundamental theorem of Linear Programming)

Given a LP in standard form, where A is a $m \times n$ matrix of rank m:

- *If there is a feasible solution, then there is a basic feasible solution.*
- *If there is an optimal feasible solution, then there is a basic optimal feasible solution.*

Remark That is, feasibility must lead to basic feasibility.

Proof [1] A feasible solution $\Longleftrightarrow$ constraint set is not empty $\Longleftrightarrow$ polytope is not empty, thus theorem also can be interpreted as feasibility must lead to basic feasibility. Let x be a feasible solution, then $Ax = b$ and $x \geq 0$. That is, $a_1 x_1 + \cdots + a_n x_n = b$ to $a_1 x_1 + \cdots + a_k x_k = b$ (some of x_i is zero). There are two possible cases:

1. $a_1, ..., a_k$ are LIN
2. $a_1, ..., a_k$ are not LIN

(i) If $k = m$, then x is a basic solution, Done! If $k < m$, since $A_{m \times n}$ is full rank, then $a_1 x_1 + \cdots + a_k x_k + a_{k+1} 0 + \cdots + a_m \cdot 0 = b$ is a basic solution, Done!

(ii) We can find the following equations, let (2)-ε(1)=$a_1 (x_1 - \varepsilon y_1) + \cdots + a_k (x_k - \varepsilon y_k) = b$, where $\varepsilon > 0$. And this is another solution to this LP, As ε increases, some of $x_i - \varepsilon y_i$ go down to zero. Repete it, we can get LIN $a_1', \cdots, a_k'$. Note that y_i can be positive or negative, however, $x_i - \varepsilon y_i$ must be positive when ε is small enough. And some of $x_i - \varepsilon y_i$ goes closer to 0 when ε increases.

$$\begin{cases} a_1 y_1 + \cdots + a_k y_k = 0 & (1) \\ a_1 x_1 + \cdots + a_k x_k = b & (2) \end{cases}$$

■

Proof [2] This is equal to show that if x is optimal, then $x - \varepsilon y$ is optimal. When ε is small, $x - \varepsilon y > 0$ then feasible, $c^\top (x - \varepsilon y) = c^\top x - \varepsilon c^\top y < c^\top x$ if $\sum c^\top y > 0$ (we can choose the sign of ε arbitrarily). Since x is optimal feasible solution, $c^T x$ is the minimal, $c^T y$ must equal to zero. Then $x - \varepsilon y$ is a optimal solution too, and by choosing ε we can get a optimal basic solution. ■

1.4.3 Basic Feasible Solution and Extreme Point

Theorem 1.4 (Extreme Point = Basic feasible solution)

Let K be the convex polytope of $H = \{Ax = b : x \in R^n, x \geqslant 0\}$. A vector x is an extreme point of K iff x is a basic feasible solution to H.

Proof If side, to show, if x is a basic feasible solution, then x is an extreme point. x is feasible $\iff x = (x_1, \ldots, x_m, 0, \ldots, 0)$ where $x_i \geq 0$ and $a_1x_1 + \cdots + a_mx_m = b$. Suppose $y = (y_1, \ldots, y_m, ; y_{m+1}, \ldots, y_n), z = (n_1, \cdots, z_m; z_{m+1}, \ldots, z_n)$ are other solutions in H. Suppose for the sake of contradiction that $\exists \alpha \in (0,1), \alpha y + (1-\alpha)z = x$, i.e., x can be represented by y and z. Then we have $\alpha y_i + (1-\alpha)z_i = 0$ for all $i = m+1, ..., n$, it means $y_i = z_i = 0 \forall i = m+1, ..., n$. Since $a_1, ..., a_m$ are LIN, there is only one kind of representation, $x_i = y_i = z_i, i = 1, ..., m$. Done!

Only if side: Say x is a extreme point. Then (1) $a_1x_1 + \cdots + a_kx_k = b$ (since x is a feasible solution). We want to prove that $a_1, ..., a_k$ are LIN. Assume that $a_1, ..., a_k$ are not LIN. We can find (2) $a_1y_1 + \cdots + a_ky_k = 0$. Construct the following equations, then $x = \frac{1}{2}(\hat{x} + \bar{x})$, that is, x is not an extreme point. Thus $a_1, ..., a_k$ must be LIN. Done!

$$\begin{cases} \hat{x} = x + \varepsilon y & (1) + \varepsilon(2) \\ \overline{x} = x - \varepsilon y & (1) - \varepsilon(2) \end{cases}$$

■

Corollary 1.1 (Nonempty Standard LP always has an extreme point)

If the convex set K corresponding to $\{Ax = b, x \geqslant 0\}$ is non-empty, then it has at least one extreme point.

Remark However, this does not mean that every nonempty polyhedron has at least one extreme point, e.g., the half space $\{(x, y) \in \mathbb{R}^2 \mid x + y \geq 1\}$.

Corollary 1.2 (Optimality and Extreme Point)

If there is a feasible solution that is optimal to a LP, then there is an optimal finite solution that is an extreme point of the constraint set.

Note on *Note that this corollary does not clarify that optimal solution is exactly the extreme point, since this optimal solution may be at the middle of the optimal line.*

Corollary 1.3 (Feasible Region and Finite Extreme Point)

The constraint set K corresponding to $\{Ax = b, x \geqslant 0\}$ has at most a finite number of extreme points.

Note on Finite *From the perspective of combination, there are at most* $\begin{pmatrix} n \\ m \end{pmatrix} = \frac{n!}{m!(n-m)!}$ *extreme points.*

Note on *These three corollary connects solution and extreme point via constraint set.*

1.4.4 Degeneracy

Definition 1.5 (Degenerate)

A basic solution $x \in R^n$ is said to be degenerate if more than n of the constraints are active at x, e.g. $a_i x = b_i$.

Definition 1.6 (Degeneracy in Standard form)

Consider the standard form polyhedron $P = \{\mathrm{x} \in \Re^n \mid \mathbf{A}\mathbf{x} = \mathbf{b}, x \geq 0\}$ and let x be a basic solution. Let m be the number of rows of A. The vector x is a degenerate basic solution if more than $n - m$ of the components of x are zero.

Lemma 1.2

If two different bases lead to the same basic solution, then this basic solution is degenerate, but not vice versa.

Proof Assume not degenerate, then the basic solution have $n - m$ zero components, this uniquely determine m non-zero components, which correspond to a unique choice of basis. Contradiction.

Counterexample is $\{(x, y) \in \mathbb{R}^2 \mid x + y \geq 0, x - y \geq 0 \quad x, y \geq 0\}$, this polyhedron contains only one degenerate point $(0, 0)$, but there is only one choice of basis. ■

Lemma 1.3

For degenerate solution, there is possible for non-basic variable's reduced cost to be negative, while it is still a optimal solution.

Proof Counterexample is $\{(x, y) \in \mathbb{R}^2 \mid x + y \geq 0, x - y \geq 0 \quad x, y \geq 0\}$. ■

1.5 Simplex Method

Assumption 1.1 (Non-degeneracy Assumption)

Every basic feasible solution is not degenerate, that is, $x_i > 0$, $i = 1, \ldots, m$ for $(x_1, \ldots, x_m, 0, \ldots, 0)$.

1.5.1 Pivoting: From basis to basis

Definition 1.7 (Pivoting)

Pivoting $\Leftrightarrow$ basis change $\Leftrightarrow$ one extreme point to another $\Leftrightarrow$ one basic solution to another. Note that the basic solution obtained by pivoting may not be feasible (may negative).

Suppose we have a modified A' in this stage, where y_{i0} is b, and we want to do pivoting based on A'.

$$A' = \begin{pmatrix} 1 & 0 & \dots & 0 & y_{1,m+1} & \dots & y_{1,n} & y_{1,0} \\ 0 & 1 & \dots & 0 & y_{2,m+1} & \dots & y_{2,n} & y_{2,0} \\ \dots & & \dots & & \dots & & \dots & \\ 0 & 0 & \dots & 1 & y_{m,m+1} & \dots & y_{m,n} & y_{m,0} \end{pmatrix}$$

Interpretation of Pivoting From the Perspective of Row: Suppose we want to use x_q $(m+1 \le q \le n)$ to replace x_p $(1 \le p \le m)$, pivoting is exactly

- Row p divided by $y_{p,q}$ (only when $y_{p,q} \neq 0$).
- Rows except Row p minus Row p and times $y_{i,q}$:

$$y'_{ij} = y_{ij} - \frac{y_{pj}}{y_{pq}} y_{iq}, \quad i \neq p; \quad y'_{pj} = \frac{y_{pj}}{y_{pq}}$$

Interpretation of Pivoting From the Perspective of Column: The polytope $\{Ax = b, x \geqslant 0\}$ can be perceived as a linear combination of $a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$. As for the canonical form, $a_j = y_{1j}a_1 + \cdots + y_{nj}a_n$ $\forall j = m+1, \dots, n$ and $b = y_{10}a_1 + y_{20}a_2 + \cdots + y_{m0}a_m$. Suppose we want to use x_q $(m+1 \le q \le n)$ to replace x_p $(1 \le p \le m)$, pivoting is exactly

- With $a_q = y_{pq}a_p + \sum_{i=1, i\neq p}^{m} y_{iq}a_i$, solve for a_p, we know $a_p = \frac{1}{y_{pq}}a_q - \sum_{i=1,i\neq p}^{m} \frac{y_{iq}}{y_{pq}}a_i$.
- By substitution with a_p, we know $a_j = \frac{y_{pj}}{y_{pq}}a_q + \sum_{i=1,i\neq p}^{m} \left(y_{ij} - \frac{y_{iq}}{y_{pq}}y_{pj}\right) a_i, \quad j = m+1, \dots, n, j \neq q$.
- That is, we do a transformation as follows

$$\begin{cases} y'_{ij} = y_{ij} - \frac{y_{iq}}{y_{pq}} y_{pj}, i \neq p \\ y'_{pj} = \frac{y_{pj}}{y_{pq}} \end{cases}$$

1.5.2 Entering basic variable

There are two ways to select the entering variable, 1st way is selecting the variable with the most negative reduced cost. However, 2nd way may be a better criterion, if we select the variable which, when pivoted in, will produce the greatest improvement in the objective function, that is, select the variable x_k corresponding to the index k that minimizes $\underset{i, y_{k,i}>0}{\text{Max}} \{r_k \cdot y_{0,i}/y_{k,i}\}$, here $\mathbf{x}_B = \mathbf{y}_0 = \mathbf{B}^{-1}\mathbf{b}$ is the current basic solution, $\mathbf{r}^T = \mathbf{c}^T - \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{A}$ is the reduced cost vector, and $\mathbf{y}_k = \mathbf{B}^{-1}\mathbf{a}_k$ where a_k is the k^{th} column in A.

Proof [2nd way] Note that $z = z_0 + \sum_{i=m+1}^{n} r_i \cdot x_i$, and to min $\sum_{i=m+1}^{n} r_i \cdot x_i$, since we can only choose one, it is equal to min $\{r_i x_i\}$.

To maintain the feasibility, assume we want to increase x_k from 0 to ε for any $k = m+1, ..., n$, and we should hold $a_1(x_1 - \varepsilon y_{ik}) + \cdots + a_m(x_m - \varepsilon y_{ik}) + a_k\varepsilon = b$ and $x_i - \varepsilon y_{ik} \geqslant 0$ $\forall i = 1, ..., m$. Thus $\varepsilon = \min_{i=1,\dots,m}\{y_{i0}/y_{ik}\}$. And the whole problem is equal to $\min_k\{r_i \min_i \{y_{i0}/y_{ik}\}\}$, since $r_i < 0$, we can rewrite as $\min_{k=m+1,\dots,n} \max_i\{r_i \frac{y_{i0}}{y_{ik}}\}$. ■

1.5.3 Leaving basic variable

Suppose the basic solution is non-degenerate and $x_i > 0, \quad i = 1, \ldots, m$, if we choose $a_q, q > m$ to enter the basis, let $\varepsilon > 0$, then we have

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$$

$$\Downarrow -\varepsilon \cdot (y_{1q}a_1 + y_{2q}a_2 + \cdots + y_{mq}a_m - a_q = 0) \tag{1.8}$$

$$a_1(x_1 - \varepsilon y_{1q}) + a_2(x_2 - \varepsilon y_{2q}) + \cdots + a_m(x_m - \varepsilon_{mq}) + \varepsilon a_q = b$$

Note that $(x_i - \varepsilon y_{iq})$ is a feasible solution as long as all $x_i - \varepsilon y_{iq} > 0$, but not a basic solution.

- If all $y_{iq} < 0$, then as ε increase, all $(x_i - \varepsilon y_{ia}) > 0$, this is a unbounded LP problem.
- If some $y_{iq} > 0$, as ε increase, we have $x_i - \varepsilon y_{iq} = 0$ for this variable, and we get a new basic solution. Let $\varepsilon_M = \min_i \left(\frac{x_i}{y_{iq}} : y_{iq} > 0\right\}$, and the corresponding vector is the one we want.
- If more than one coefficient reduces to zero at $\varepsilon = \varepsilon_M$, the new basic solution is degenerate.

1.5.4 Optimality Test

Given a basic feasible solution $x = (x_1, \ldots, x_m, 0, \ldots, 0)$, let $z_0 = \sum_{i=1}^{m} c_i y_{i0}$ (i.e., current objective). Then the objective function can be rewritten as the type contains z_0, and whether there exists pivoting to optimize $\sum_{i=m+1}^{n}(c_i - z_i)x_i$ means whether the current solution is optimal.

$$\begin{aligned}
z &= \sum_{i=1}^{n} c_i x_i \\
&= \sum_{i=1}^{m} c_i x_i + \sum_{i=m+1}^{n} c_i x_i \\
&= \sum_{i=1}^{m} c_i \Big(y_{i0} - \sum_{j=m+1}^{n} y_{ij} x_j\Big) + \sum_{i=m+1}^{n} c_i x_i && \Big(x_i = y_{i0} - \sum_{j=m+1}^{n} y_{ij} x_j\Big) \\
&= z_0 + \sum_{i=m+1}^{n} (c_i - z_i) x_i && \Big(z_i = \sum_{k=1}^{m} y_{kj} c_k\Big)
\end{aligned}$$

Theorem 1.5 (Optimality Condition)

Given a non-degenerate basic feasible solution with corresponding objective function value z_0:

- *If $c_i - z_i < 0$ for some i, then there is a feasible solution with objective value $z < z_0$. If the column a_i can be substituted for some column in the original basis to yield a new basic feasible solution, then this new solution will have $z < z_0$. Otherwise if we found a vector d satisfying $Ad = 0, d \geq 0, c'd < 0$ (Bertsimas et al., 1997, P. 91), the constraint set is unbounded and the objective function value can be made arbitrarily small.*
- *If $c_i - z_i \geq 0$ for all i, then the solution is optimal.*

Lemma 1.4 (ε-optimal)

For a standard LP, say if $|z_0 - z^*| \le \varepsilon$, then it would be enough, here z_0 is the current value of simplex. Let $\sum_i x_i \le s$, then if $M = \max_j(z_j - c_j) \le \varepsilon/s$, then $z_0 - z^* \le \varepsilon$.

Proof Note that

$$
\begin{aligned}
|z - z_0| &= |\sum_{i=m+1}^{n} (c_i - z_i)\, x_i| \\
&\le \sum_{i=m+1}^{n} |c_i - z_i| x_i && |\sum_i x_i| \le \sum_i |x_i| \\
&\le M \sum_{i=m+1}^{n} x_i && z_j - c_j \le M \\
&\le Ms \le \varepsilon
\end{aligned}
$$

■

1.5.5 Matrix formulation of Simplex Method

Let $A = (B|D)$, $x = (x_B|x_D)$, and $c^T = (c_B^T, c_D^T)$. Formulating LP with matrix,

$$
\begin{aligned}
\text{Min} \quad & z = c_B^T x_B + C_D^T x_D \\
\text{s.t.} \quad & Bx_B + Dx_D = b \quad \Rightarrow \quad x_B = B^{-1}b - B^{-1}Dx_D \\
& x_B, x_D \geqslant 0
\end{aligned}
\tag{1.9}
$$

We have $z = c_B^\top B^{-1}b + \left(c_D{}^\top - C_B^\top B^{-1}D\right) x_D$, where $r_D{}^\top := c_D{}^\top - c_B{}^\top B^{-1}D$ is the reduced cost vector. We also have a basic solution $\left(x_B = B^{-1}b, x_D = 0\right)$.

- Initaialization: Given the basis B, the current solution is $B^{-1}b$.
- Step 1: Calculate the relative cost vector $r_D^T = c_D^T - c_B^T B^{-1}D$. If $r_D^T \ge 0$, the current basis is optimal. Otherwise, go to Step 2.
- Step 2: Determine which vector to enter the basis by selecting the most negative cost coefficient. Let it be column q and then $B^{-1}a_q$ gives the representation of a_q in terms of the vectors in the current basis B.
- Step 3: If all $y_{iq} \le 0$, then stop and the problem is unbounded. Otherwise, calculate the ratio of $\frac{y_{i,0}}{y_{i,q}}$ for $y_{i,q} > 0$, and determine which variable to enter the basis.
- Step 4: Update B and the current solution $B^{-1}b$. Return to Step 1.

1.5.6 Simplex and Degeneracy

1.5.7 Modified Simplex Method for LP with Bounded Variables

- Given a feasible solution x^0.
- choose the entering variable be $s = \arg\min_{\{j\in L\}\cup\{k\in U\}}\{r_j, -r_k\}$, and define δ. We will

do the change to make x_s to $x_S = x_S^0 + \delta\theta, \theta \geq 0$.

$$\delta = \begin{cases} 1 & \text{if } x_S = l_S \\ -1 & \text{if } x_S = u_S \end{cases}$$

- choose the leaving variable: To maintain feasilibity, we need $l_S \leq x_S^0 + \delta\theta \leq u_S$ and $l_i \leq x_i^0 - \delta\theta y_{iS} \leq u_i, i = 1, 2 \ldots, m$, thus, $\theta = \min\{\theta_S, \theta_l, \theta_u\}$. Let $r = \operatorname{argmin}\{\theta_S, \theta_l, \theta_u\}$, x_r is the leaving variable we want.

$$\theta_S = u_S - l_S, \theta_l = \min_{\{i|\delta y_{iS}>0\}} \left\{\frac{x_i^0 - l_i}{\delta y_{is}}\right\} \geq 0, \theta_u = \min_{\{i|\delta y_{iS}<0\}} \left\{\frac{u_i - x_i^0}{-\delta y_{is}}\right\} \geq 0$$

1.6 Artificial Variable

Artificial variable is used to find an initial basic solution.

1.6.1 Big M

Lemma 1.5 (Lower bound for M)

A finite value for such an M must exist, and it is $\max\{c_B^T B^{-1}\}$, here B is the optimal basis for the primal problem.

Proof Suppose B is the optimal basis for the primal problem and x^* is the corresponding optimal solution, it is equal to discuss the problem of introducing multiple new variables and keep the optimality. Thus we need the reduced cost for these new variables being non-negative, that is, $r_{x_a} = M - c_B^T B^{-1} I \geq 0$. ■

1.6.2 Two phase

1.7 Transportation Problem

$$\begin{array}{llll}
 & \textbf{Primal} & & \textbf{Dual} \\
\min & \sum_{i,j} c_{ij} x_{ij} & \max & \sum_{i=1}^{m} a_i u_i + \sum_{j=1}^{n} b_j v_j \\
\text{s.t.} & \sum_{j=1}^{n} x_{ij} = a_i \quad \text{for} \quad i = 1, \ldots, m & \text{s.t.} & u_i + v_j \leqslant c_{ij}, \quad \forall v, j \\
 & \sum_{i=1}^{m} x_{ij} = b_j \quad \text{for} \quad j = 1, \ldots, n & & \\
 & x_{ij} \geqslant 0 \quad \forall v, j & &
\end{array} \tag{1.10}$$

Chapter 2 Large Scale Linear Programming

2.1 Dantzig-Wolfe Decomposition

2.2 Column Generation

2.3 Row Generation

Chapter 3 Convex Optimization

Chapter 4 Duality

4.1 Weak, Strong Duality Theorem

4.1.1 Duality Form

Definition 4.1 (Symmetric Duality Form)

$$\begin{array}{llll} & \textbf{\textit{Primal}} & & \textbf{\textit{Dual}} \\ \min & c^T x & \max & \lambda^T b \\ \textit{s.t.} & Ax \geq b, x \geq 0 & \textit{s.t.} & \lambda^T A \leq c^T, \lambda \geq 0 \end{array} \tag{4.1}$$

Definition 4.2 (Asymmetric Duality Form)

$$\begin{array}{llll} & \textbf{\textit{Primal}} & & \textbf{\textit{Dual}} \\ \min & c^T x & \max & \lambda^T b \\ \textit{s.t.} & Ax = b, x \geq 0 & \textit{s.t.} & \lambda^T A \leq c^T \end{array} \tag{4.2}$$

Proof [Asymmetric is a special case of Symmetric]

$$\begin{array}{llll} & \textbf{Primal} & & \textbf{Dual} \\ \min & c^T x & \max & \lambda_1^T b - \lambda_2^T b = (\lambda_1^T - \lambda_2^T) b \\ \text{s.t.} & Ax \geq b \quad (\lambda_1), x \geq 0 & \text{s.t.} & \lambda_1^T A - \lambda_2^T A = (\lambda_1^T - \lambda_2^T) A \leq c^T \\ & -Ax \geq -b \quad (\lambda_2) & & \lambda_1 \geq 0, \lambda_2 \geq 0 \end{array} \tag{4.3}$$

Just let $\lambda = (\lambda_1^T - \lambda_2^T)$ then we prove it. ■

Note on Table of Duality Transformation

Dual(Max)	Primal(Min)
ith const $\leq$	ith var ≥ 0
ith const $=$	ith var unrestricted
jth var $\geq$	jth const ≥ 0
jth var unrestricted	jth const $= 0$

Definition 4.3 (Duality gap)

The gap between primal objective and dual objective.

Remark[LP's Duality gap] Duality gap for LP is zero.

Example 4.1Bounded variable LP's Dual

Primal	Primal 2	Dual	Dual 2
$\min \quad c^T x$	$\min \quad c^T x_1 - c^T x_2$	$\max \quad l^T y_1 - u^T y_2$	$\max \quad l^T y_1 - u^T y_2$
$\text{s.t.} \quad l \le x \le u$	$\text{s.t.} \quad x_1 - x_2 \ge l$	$\text{s.t.} \quad y_1 - y_2 \le c$	$\text{s.t.} \quad y_1 - y_2 = c$
	$-(x_1 - x_2) \ge -u$	$-(y_1 - y_2) \le -c$	
	$x_1, x_2 \ge 0$	$y_1, y_2 \ge 0$	$y_1, y_2 \ge 0$

Example 4.2Primal's variant to Dual Consider the primal LP, suppose primal and dual are feasible, let λ be a known optimal solution to the dual.

$$
\begin{aligned}
\text{Minimize} \quad & \boldsymbol{c}^T \boldsymbol{x} \\
\text{Subject to} \quad & \boldsymbol{A}\boldsymbol{x} \ge \boldsymbol{b}, \quad \boldsymbol{A} : m \times n \\
& \boldsymbol{x} \ge \boldsymbol{0}.
\end{aligned}
$$

1. If the kth equation of the primal is multiplied by $\mu \neq 0$, an optimal solution w to the dual of this new problem should be: On the basis of $\mu a_k \lambda_k = a_k w_k$ and $\mu c_k \lambda_k = c_k w_k$, we have $w_k = \frac{\lambda_k}{\mu}, w_{i \neq k} = \lambda_i$.
2. If we add μ times the kth equation to the rth equation, an optimal solution w to the dual of this new problem should be: On the basis of $b_k \lambda_k + b_r \lambda_r = b_k w_k + (\mu b_k + b_r) w_r$ and $a_k \lambda_k + a_r \lambda_r = a_k w_k + (\mu a_k + a_r) w_r$, we have $w_k = \lambda_k - \mu \lambda_r, w_{i \neq k} = \lambda_i$.
3. If we add μ times the kth equation to c, an optimal solution w to the dual of this new problem should be: Suppose w is the same as λ except kth element, based on $\sum_{i=1}^m a^i w_i = \sum_{i=1}^m a^i \lambda_i + a^k w_k - a^k \lambda_k \le (c^T + \mu a^k)$, recalling that $\sum_{i=1}^m a^i \lambda_i \le c^T$ and if $a^k(w_k - \lambda_k) = \mu a^k$, then w is feasible too. Here $w_k = \lambda_k + \mu, w_{i \neq k} = \lambda_i$. And notice that $w^T b = \sum_{i=1, i \neq k}^m \lambda_i + (\lambda_k + \mu) b_k = (c^T + \mu a^k) x^0$, thus w is optimal too.

$$
\begin{array}{llll}
 & \textbf{Primal} & & \textbf{Dual} \\
\min & (c^T + \mu a^k) x & \max & w^T b \\
\text{s.t.} & Ax = b, x \ge 0 & \text{s.t.} & w^T A \le (c^T + \mu a^k)
\end{array}
\tag{4.4}
$$

4. If the RHS changes from b to b', the resulting program is infeasible or has a finite optimal feasible solution. Since the dual feasibility does not change, and the dual problem is still feasible, so the primal problem should be infeasible or finite optimal.

$$
\begin{array}{llll}
 & \textbf{Primal} & & \textbf{Dual} \\
\min & c^T x & \max & w^T b' \\
\text{s.t.} & Ax = b', x \ge 0 & \text{s.t.} & w^T A \le c^T, \lambda \ge 0
\end{array}
\tag{4.5}
$$

4.1.2 Clark's Theorem

Lemma 4.1 (Clark's Theorem)

Given the following primal and dual LPs, if one of them is feasible, then the feasible region for one of them is non-empty and unbounded.

$$\begin{array}{llll} & \textbf{\textit{Primal}} & & \textbf{\textit{Dual}} \\ \min & c^T x & \max & \lambda^T b \\ s.t. & Ax \geq b, x \geq 0 & s.t. & \lambda^T A \leq c^T, \lambda \geq 0 \end{array} \tag{4.6}$$

Remark It's important to note that the result of the theorem is that the feasible region of one of the LPs is unbounded, but it may not be the case that the LP has unbounded objective function value with the given objective function.

Proof There are three possibilities to consider.

1. The primal is infeasible and the dual is unbounded. Done!
2. The dual is infeasible and the primal is unbounded. Done!
3. Both the primal and the dual are finite optimal.

Suppose both the primal and the dual are finite optimal, let $\hat{c} = [-1, -1, \ldots, -1]$ and consider the following systems:

1. $\exists \hat{y}$ such that $A^T \hat{y} \leq \hat{c}, \hat{y} \geq 0$
2. $\exists \hat{x}$ such that $A\hat{x} \geq 0, \hat{c}^T \hat{x} < 0, \hat{x} \geq 0$

Farkas' lemma tells us exact one of them holds.

1. If (2) holds, then $\hat{x}$ is a feasible solution to primal LP and suppose x is also feasible to primal LP too, note that $\hat{x} \neq 0$ because $\hat{c}^T \hat{x} < 0$, for any $\lambda > 0$, we have another feasible $x + \lambda\hat{x}$ for primal too. By enlarge λ, we have a unbounded feasible region for primal LP.

$$A(x + \lambda\hat{x}) = Ax + \lambda A\hat{x} \geq b + \lambda * 0 = b$$

2. If (1) holds, then $\hat{y}$ is a feasible solution to dual LP ($\hat{y}$ is not 0 since $A^T \hat{y} \leq \hat{c}$), and suppose y is also feasible to dual LP too. Similarly, we have $y + \lambda\hat{y}$ is feasible to dual LP for any $\lambda > 0$.

■

4.1.3 Weak Duality Theorem

Theorem 4.1 (Weak Duality Theorem)

Let x and λ be the feasible solutions to the Primal and Dual respectively. Then $\lambda^T b \leq c^T x$.

Proof By the feasible conditions $Ax \geq b$ and $\lambda^T A \leq c^T$, we have $\lambda^\top b \leqslant \lambda^\top Ax \leqslant c^\top x$. ■

Note on *In the case of LP, the dual gap is always zero, while this is not true in other optimization problem. Weak duality theorem points out the lower bound of the primary problem, $\lambda^\top b \leqslant \lambda^\top Ax \leqslant c^\top x$ as long as we can find a λ such that $\lambda^T A \leq c^T$.*

Corollary 4.1 (Equal Primal-Dual Feasible means Optimal)

If x_0 and λ_0 are feasible to the Primal and Dual respectively and if $\lambda_0^T b = c^T x_0$, then x_0 and λ_0 are optimal to their respective problems.

Proof Assume that x_0 and λ_0 are not optimal, we can find that $c^T x_0 = \lambda_0^T b < \lambda_1^T b$, and this contradicts the Weak Duality Theorem. ■

4.1.4 Strong Duality Theorem

Theorem 4.2 (Strong Duality Theorem for LP)

If either the Primal or the Dual has a finite optimal solution, so does the other; the corresponding values of the objective are equal. If either problem has an unbounded objective, the other problem has no feasible solution.

Corollary 4.2 (Optimal Primal to Optimal Dual)

Let the Primal problem have an optimal basic feasible solution corresponding to the basis B. Then the vector λ satisfying $\lambda^T = c_B^T B^{-1}$ is an optimal solution to the Dual. The optimal solutions to both program are equal.

Proof If we partition A as $A = (B|D)$, and assume the optimal basis is B, then $x_B = B^{-1}b$, and the optimal value is ${C_B}^\top B^{-1}b$. The reduced cost vector is $r^T = (r_B|r_D)^T$, and ${r_D}^\top = {C_D}^\top - {C_B}^\top B^{-1}D \geqslant 0$, $r_B^\top = C_B^\top - C_B^\top B^{-1}B = 0$. Thus $C_B^\top B^{-1}D \leqslant C_D T^\top$, this means that λ^T is a feasible solution for Dual. And $\lambda^\top b = C_B^\top B^{-1}b = C_B^T x_B$, by weak duality theorem we know it is optimal. ■

Note on *Farkas lemma can be used to prove strong duality theorem and can also be proved by strong duality theorem (Ali Ahmadi, 2016, Lec. 5).*

Proof [Strong Duality to Farkas Lemma] Easy to see both condition can not holds simultaneously, then we prove if not (1) then (2).

$$\begin{array}{llll} & \textbf{Primal} & & \textbf{Dual} \\ \min & 0 & \max & \lambda^T b \\ \text{s.t.} & Ax = b, x \geq 0 & \text{s.t.} & \lambda^T A \leq 0 \end{array} \tag{4.7}$$

Here we prove if primal is infeasible then dual is unbounded. Easy to see dual must be feasible ($\lambda = 0$), then dual is unbounded, which means there exists λ such that $\lambda^T A \leq 0, \lambda^T b > 0$. ■

4.1.5 Dual solution from primal simplex table

Below is an example of how to obtain the dual solution directly from the final simplex tableau of the primal.

$$
\begin{array}{llll}
 & \textbf{Primal} & & \textbf{Dual} \\
\min & -x_1-4x_2-3x_3 & \max & 4\lambda_1+6\lambda_2 \\
\text{s.t.} & 2x_1+2x_2+x_3\leq 4 & \text{s.t.} & 2\lambda_1+\lambda_2\leqslant -1 \\
 & x_1+2x_2+2x_3\leqslant 6 & & 2\lambda_1+2\lambda_2\leqslant -4 \\
 & x_1,x_2,x_3\geqslant 0 & & \lambda_1+2\lambda_2\leqslant -3 \quad \lambda_1,\lambda_2\leqslant 0
\end{array}
\tag{4.8}
$$

$$
\begin{array}{ccccccc}
\frac{3}{2} & 1 & 0 & 1 & -\frac{1}{2} & 1 & \lambda_1 \\
-1 & 0 & 1 & -1 & 1 & 2 & \lambda_2 \\
2 & 0 & 0 & 1 & 1 & 10 &
\end{array}
$$

Here the 1st and 2nd row correspond to λ_1 and λ_2, the 4th and 5th column are slack variables, the 2nd and 3rd column are basic variables. By $r^T = C_D^\top - C_B^\top B^{-1}D$, we know that for s_1, we have $0-(\lambda_1,\lambda_2)\begin{pmatrix}1\\0\end{pmatrix} = 1 \Rightarrow \lambda_1=-1$, and for s_2 we have $0-(\lambda_1,\lambda_2)\begin{pmatrix}0\\1\end{pmatrix} = 1 \Rightarrow \lambda_2=-1$.

4.1.6 Homogenous form of Dual

Theorem 4.3

$$
\begin{array}{rcrrcr}
\textit{Primal} & & \textit{Dual} & \textit{HD} & & \textit{PD} \\
\textit{finite optimal} & \rightleftarrows & \textit{finite optimal} & & & \\
\textit{unbounded} & \not\rightleftarrows & \textit{infeasible} & & & \\
\textit{infeasible} & \not\leftrightarrows & \textit{unbounded} & & & \\
\textit{infeasible} & \rightarrow & & \textit{unbounded} & \rightarrow & \textit{infeasible} \\
 & & & \textit{finite optimal} & \rightarrow & \textit{finite optimal}
\end{array}
\tag{4.9}
$$

We can construct the homogenous form of dual problem as follows.

$$
\begin{array}{llllllll}
 & \textit{Primal} & & \textit{Dual} & & \textit{HD} & & \textit{HD's Dual} \\
\min & c^Tx & \max & \lambda^Tb & \max & \lambda^Tb & \min & 0 \\
\textit{s.t.} & Ax\geq b & \textit{s.t.} & \lambda^TA\leq c^T & \textit{s.t.} & \lambda^TA\leq 0 & \textit{s.t.} & Ax\geq b \\
 & x\geq 0 & & \lambda\geq 0 & & \lambda\geq 0 & & x\geq 0
\end{array}
\tag{4.10}
$$

*Homogenous form has a nice property: it must be feasible, e.g. $\lambda=0$. Note that **PD** and **P** has the same feasible region, thus they have the same feasibility.*

Lemma 4.2 (Unbounded condition)

Suppose the following LP problem is feasible:

$$\begin{aligned} \textit{Minimize} \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \textit{Subject to} \quad & \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}, \quad \boldsymbol{A} : m \times n \\ & \boldsymbol{x} \geq \boldsymbol{0}. \end{aligned}$$

The optimal solution approaches to $-\infty$ if and only if there exists an $\bar{x} \neq 0$ such that $\overline{\boldsymbol{x}} \geq \boldsymbol{0}, \boldsymbol{A}\overline{\boldsymbol{x}} \geq \boldsymbol{0}, \boldsymbol{c}^T\overline{\boldsymbol{x}} < \boldsymbol{0}$.

Proof

$$\begin{array}{llllllll} & \textbf{Primal} & & \textbf{Dual} & & \textbf{HP's Dual} & & \textbf{HP} \\ \min & c^T x & \max & \lambda^T b & \max & 0 & \min & c^T x \\ \text{s.t.} & Ax \geq b & \text{s.t.} & \lambda^T A \leq c^T & \text{s.t.} & \lambda^T A \leq c^T & \text{s.t.} & Ax \geq 0 \\ & x \geq 0 & & \lambda \geq 0 & & \lambda \geq 0 & & x \geq 0 \end{array} \tag{4.11}$$

If side: since $\bar{x}$ is a feasible solution to (HP), thus (HP) is feasible (unbounded or finite). Assume (HP) is finite optimal, so thus (HP's Dual), and $c^T\bar{x} < 0$ contradicts the weak duality theorem, thus the assumption is wrong, (HP) is unbounded and (HP's Dual) is infeasible. (HP's Dual) and (Dual) share the same feasible region, then (Dual) is infeasible too. Since (Primal) is feasible, then it must be unbounded.

If side (2): Suppose there is a feasible solution x^*, then $x^* + \lambda\bar{x}$ is also feasible ($\lambda \geq 0$), and we can increase λ to infinity and the optimal value is negative infinity.

Only if side: (Primal)'s unbounded means (Dual) is infeasible and also (HP's Dual), and (HP) must be feasible (0 is a feasible solution), thus (HP) is unbounded too. It means we can find a solution, which is feasible ($\bar{x} \geq 0, A\bar{x} \geq 0$) and $c^T\bar{x} < 0$ (unbounded and min objective function). ■

4.1.7 Complementary Slackness

Theorem 4.4 (Complementary Slackness– Asymmetric Form)

Let x and λ be feasible solutions for the primal and dual programs, respectively. A necessary and sufficient condition that they both be optimal solutions is that for all i

- $x_i > 0 \Rightarrow \mathbf{y}^T\boldsymbol{a}_i = c_i$
- $x_i = 0 \Leftarrow \mathbf{y}^T\boldsymbol{a}_j < c_j$

Proof Note that in both side we have $(y^T A - c^T)x = 0$. ■

Theorem 4.5 (Complementary Slackness– Symmetric Form)

Let x and λ be feasible solutions for the primal and dual programs, respectively. A necessary and sufficient condition that they both be optimal solutions is that for all i and j (where a^j is the jth row of A)

- $x_i > 0 \Rightarrow \mathbf{y}^T \mathbf{a}_i = c_i$
- $x_i = 0 \Leftarrow \mathbf{y}^T \mathbf{a}_i < c_i$
- $\lambda_j > 0 \Rightarrow \mathbf{a}^j \mathbf{x} = b_j$
- $\lambda_j = 0 \Leftarrow \mathbf{a}^j \mathbf{x} > b_j$

Lemma 4.3 (Primal-Dual Feasible + Complementary Slackness = Optimal)

Given a primal-feasible solution x and a dual-feasible solution y, x and y are optimal iff the complementary slackness conditions hold.

Example 4.3 Consider the following LP (P. Williamson, 2014, PS. 1), assume that v_i, s_i are positive and $\frac{v_1}{s_1} \geq \frac{v_2}{s_2} \geq \cdots \geq \frac{v_n}{s_n}$, let k be the largest index such that $s_1 + s_2 + \cdots + s_{k-1} \leq B$. Find a optimal solution to primal and dual.

$$
\begin{array}{rlrl}
 & \textbf{Primal} & & \textbf{Dual} \\
\max & \sum_{i=1}^{n} v_i x_i & \min & \sum_{i=1}^{n} y_i + B y_0 \\
\text{s.t.} & \sum_{i=1}^{n} s_i x_i \leq B & \text{s.t.} & [s \quad I^T](y_0, ..., y_n)^T \geq v \\
 & x_i \leq 1 \quad i = 1, \ldots, n & & y_i \geq 0 \quad i = 0, ..., n \\
 & x_i \geq 0 \quad i = 1, \ldots, n & &
\end{array}
\tag{4.12}
$$

Solution *The logic is that the first $k-1$ variable contribute most value to objective, thus they must be the maximum value, that is, 1. And the kth variable can achieve maximum smaller than 1 due to the first constraint. By complementary slackness we know the dual variable behind kth must be 0, and solve the following equations we derive the dual solution.*

$$
x_i = \begin{cases} 1 & i < k \\ \frac{B-(s_1+s_2+\cdots+s_{k-1})}{0} & i = k \\ 0 & i > k \end{cases}
$$

$$
y_i = \begin{cases} \frac{v_k}{s_k} & i = 0 \\ s_i \left(\frac{v_i}{s_i} - \frac{v_k}{s_k} \right) & 0 < i < k \\ 0 & i \geq k \end{cases}
$$

4.1.8 Degeneracy and Uniqueness under Duality

1. Since λ is m-dimensional, dual degeneracy implies more than m reduced costs that are zero.
1. If dual has a nondegenerate optimal solution, the primal problem has a unique optimal solution. However, it is possible that dual has a degenerate solution and the dual has a unique optimal solution.

4.1.9 Redundant Equations (Luenberger and Ye, 2015, Ch. 4)

Definition 4.4 (Reduandant equations)

Corresponding to the system $Ax = b, x \geq 0$, we say the system has redundant equations if there is a nonzero λ satisfying $\lambda^T A = 0, \lambda^T b = 0$.

Remark This means that one of the equations can be expressed as a linear combination of the others.

Definition 4.5 (Null variable)

Corresponding to the system $Ax = b, x \geq 0$, a variable x_i is said to be a null variable if $x_i = 0$ in every solution.

Example 4.4

$$2x_1 + 3x_2 + 4x_3 + 4x_4 = 6$$
$$x_1 + x_2 + 2x_3 + x_4 = 3$$
$$x_1 \geqslant 0, x_2 \geqslant 0, x_3 \geqslant 0, x_4 \geqslant 0$$

Twice the second row minus the first row leads to $x_2 + 2x_4 = 0$. Thus, both of two variables are null variable.

Lemma 4.4 (Null value theorem)

If S is not empty, the variable x_i is a null variable in the system $Ax = b, x \geq 0$ iff there is a nonzero vector λ such that $\lambda^T A \geq 0, \lambda^T b = 0$ and the ith component of $\lambda^T A$ is strictly positive.

Definition 4.6 (Nonextremal variable)

A variable x_i in the system $Ax = b, x \geq 0$ is nonextremal if the inequality $x_i \geq 0$ is redundant.

Lemma 4.5 (Nonextremal variable theorem)

If S is not empty, the variable x_j is a nonextremal variable for the system $Ax = b, x \geq 0$ iff there is $\lambda \in E^m$ and $d \in E^n$ such that

$$\boldsymbol{\lambda}^T \mathbf{A} = \mathbf{d}^T \quad d_j = -1, \quad d_i \geqslant 0 \quad \textit{for} \quad i \neq j$$

and such that

$$\boldsymbol{\lambda}^T \mathbf{b} = -\beta \quad \textit{for some } \beta \geq 0$$

Lemma 4.6 (Inconsistent systems of linear inequalities (Bertsimas et al., 1997, P. 194))

Let $a_1, ..., a_m$ be some vectors in R^n, with $m > n + 1$. Suppose that the system of inequalities $\mathbf{a}_i'\mathbf{x} \geq b_i, i = 1, \ldots, m$, does not have any solutions. Show that we can choose $n + 1$ of these inequalities, so that the resulting system of inequalities has no solutions.

4.2 Dual Simplex Method

4.3 Shadow Price and Sensitivity Analysis

Definition 4.7 (Shadow Price)

The shadow price to a constraint i is the rate of the change in the objective function value as a result of a change in the value of b_i.

Definition 4.8 (Simplex Multiplier)

$\lambda^T = c_B^T B^{-1}$

4.3.1 Introducing a new variable

$$\begin{aligned} \text{Minimize} \quad & c^T x + c_{n+1} x_{n+1} \\ \text{Subject to} \quad & Ax + a_{n+1} x_{n+1} = b \\ & x \geq 0, x_{n+1} \geq 0 \end{aligned}$$

1. The feasibility of the solution is not affected (the feasibility region is enlarged), but the solution may not be optimal. We now have more choices to form the basis.
2. Firstly, check if $c_{n+1} - C_B^T B^{-1} a_{n+1} \geq 0$ still holds. If so, the former optimal solution is still optimal, else x_{n+1} should enter the basis and we need to find the leave variable.

4.3.2 Introducing a new constraint

Lemma 4.7 (Introducing a new constraint)

Consider the LP in standard form, assume x^0 is an optimal solution to the problem. Introducing a new constraint $\boldsymbol{a}^T \boldsymbol{x} \leq \varphi$.

$$\begin{aligned} \textit{Minimize} \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \textit{subject to} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

1. *Prove that if $\boldsymbol{a}^T \boldsymbol{x} \leq \varphi$, then x^0 is optimal for the new problem too.*
2. *Prove that if $\boldsymbol{a}^T \boldsymbol{x} > \varphi$, then either there exists no feasible solution to the original problem or there exists an optimal solution x^* such that $a^T x^* = \varphi$.*

Proof

$$\begin{array}{llllllll} & \textbf{P1} & & \textbf{D1} & & \textbf{P2} & & \textbf{D2} \\ \min & c^T x & \max & \lambda^T b & \min & c^T x & \max & \lambda^T b - \lambda_{m+1}\varphi \\ \text{s.t.} & Ax = b & \text{s.t.} & \lambda^T A \leq c^T & \text{s.t.} & Ax = b, a^T x \leq \varphi & \text{s.t.} & \lambda^T A - \lambda_{m+1} a \leq c^T \\ & x \geq 0 & & & & x \geq 0 & & \lambda_{m+1} \geq 0 \end{array} \tag{4.13}$$

If $\boldsymbol{a}^T\boldsymbol{x} \leq \varphi$, assume λ^0 is the optimal solution to (D1), then $c^Tx^0 = (\lambda^0)^Tb$. We can construct a solution $(\lambda^0, \lambda_{m+1} = 0)$ to (D2). Note that $(\lambda^0, \lambda_{m+1} = 0)$ is feasible to (D2) and $c^Tx^0 = (\lambda^0)^Tb - 0 \cdot \varphi$, thus x_0 is still the optimal solution to (P2).

$$\begin{array}{llcll} & \textbf{P3} & & & \textbf{D3} \\ \min & 0 & & \max & \lambda^T b - \lambda_{m+1}\varphi \\ \text{s.t.} & Ax = b, a^T x \leq \varphi & & \text{s.t.} & \lambda^T A - \lambda_{m+1} a \leq 0 \\ & x \geq 0 & & & \lambda_{m+1} \geq 0 \end{array} \tag{4.14}$$

If $\boldsymbol{a}^T\boldsymbol{x} > \varphi$, since 0 is a feasible solution to (D3), thus (D3) is feasible.

1. If (D3) is unbounded, then (P3) is infeasible. Since (P2) and (P3) share the same feasible region, thus (P2) is infeasible too.
2. If (D3) is finite optimal, so thus (P3) and (P2). Assume that for all optimal solution x^*, $a^Tx^* < \varphi$, and also we have $\lambda_{m+1} = 0$ according to the complementary slackness property. That is, (D2) and (D1) share the same feasible region and achieves the same optimal solution, (D2) can be rewritten as (D1). Thus (D2)'s dual problem should share the same optimal solution as (P1) too, and all optimal solution for (P1) should satisfy $a^Tx \leq \varphi$, and this contradicts our assumption. Therefore, there exists an optimal solution x^* for (P2), where $a^Tx^* = \varphi$.
3. If (D3) is finite optimal, so thus (P3) and (P2). Assume that for all optimal solution x^*, $a^Tx^* < \varphi$, and also we have $\lambda^*_{m+1} = 0$ according to the complementary slackness property. Denote the optimal solution set for (P1), (D1), (P2), (D2) as X_1, Y_1, X_2, Y_2, then for any $x_1 \in X_1, \lambda_1 \in Y_1, x_2 \in X_2, \lambda_2 \in Y_2$, $c^\top x^2 = \left((\lambda^1)^\top\right) b = (\lambda^2)^\top b = c^\top x'$. This means $X_2 \in X_1$. To prove this, assume there is $x_2 \in X_2$ and $x_2 \notin X_1$, but x_2 is feasible to (P1) and $c^Tx^2 = c^Tx^1$, contradiction. Note that the optimal solution set must be convex, and it means the hyperplane $a^Tx = \varphi$ splits the set X_1, and X_2 is the part of X_1 located in the negative half space of $a^Tx = \varphi$. Thus $X_2 \cap \{x | a^\top x = \varphi\}$ is not empty, there exists an optimal solution x^* such that $a^Tx^* = \varphi$.

■

$$\begin{array}{ll} \text{Minimize} & c^T x \\ \text{Subject to} & Ax = b \\ & a^{n+1}x = b_{n+1} \\ & x \geq 0 \end{array}$$

1. The current solution is also optimal if it satisfies the augmented constraint. Introducing a new constraint is actually introducing a new hyperplane and reduce the feasibility region.
2. Otherwise, ...

4.3.3 Change Cost coefficient for a non-basic variable

$$\begin{aligned} \text{Minimize} \quad & \textstyle\sum_{i=1, i\neq j}^{n} c_i x_i + (c_j + \Delta) x_j \\ \text{Subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

1. The feasibility $B^{-1}b$ does not change too.
2. Note that the simplex multipliers $C_B B^{-1}$ are not affected, thus the next thing is to check the reduced cost whether $r_j(\Delta) = c_j + \Delta - \lambda^T a_j = \Delta + r_j \geq 0$. If so, then the optimal remains, otherwise, this non-basic variable should enter the basis.

4.3.4 Change Cost coefficient for a basic variable

Consider a change in the cost coefficient c_i of a basic variable x_i to $c_i + \Delta$:

1. The feasibility holds.
2. The basic cost vector changes from c_B to $c_B(\Delta) = c_B + \Delta e_i$.
3. The updated simplex multipliers are $\lambda^T(\Delta) = (c_B(\Delta))^T B^{-1} = \lambda^T + \Delta e_i^T B^{-1}$.
4. The reduced cost coefficient for a non-basic variable x_j is $r_j(\Delta) = c_j - \lambda^T(\Delta) a_j = r_j - \Delta e_i^T B^{-1} a_j = r_j - \Delta y_{ij}$. Thus the range of Δ for which the current solution remains optimal is given by $\max_{y_{ij}<0} \frac{r_j}{y_{ij}} \leq \Delta \leq \min_{y_{ij}>0} \frac{r_j}{y_{ij}}$.

4.3.5 Changing RHS scalar

Consider a change in a RHS scalar b_i to $b_i + \Delta$:

1. The simplex multipliers are unaffected and the optimality condition holds.
2. If the feasibility holds, then it is still optimal. If the feasibility does not hold, then apply the Dual Simplex Method.
3. Note that $x_B(\Delta) = B^{-1}(b + \Delta e_i) = x_B + \Delta B^{-1} e_i$, thus it may not be feasible.
4. The range of Δ for which the current solution remains optimal is given by $\max_{\beta_{ki}>0} \frac{-x_{B_k}}{\beta_{ki}} \leq \Delta \leq \min_{\beta_{ki}<0} \frac{-x_{B_k}}{\beta_{ki}}$, where β_{ki} is the kith element of B^{-1}.
5. If the current solution remains optimal, the objective function value changes to $z^*(\Delta) = z^* + \Delta\lambda_i$, where λ_i is the ith element in the vector of simplex multipliers.

4.3.6 Changing a non-basic column

Consider a change in a coefficient a_{kj} in a non-basic column vector $a_j, k = 1, 2..., m; j = m+1, ..., n$ to $a_{kj} + \Delta$, that is, $a_j(\Delta) = a_j + \Delta e_k$.

1. The feasibility of the solution and simplex multipliers remain unaffected.
2. The reduced cost coefficient of x_j is $r_j(\Delta) = c_j - \lambda^T a_j(\Delta) = r_j - \Delta\lambda^T e_k = r_j - \Delta\lambda_k$.
3. The range of Δ for which the current solution remains optimal is $\Delta\lambda_k \leq r_j$. If $\lambda_k = 0$, then the optimality is not affected by row k.

4.3.7 Changing a basic column

Consider a change in a coefficient a_{ki} in a basic column vector $a_i, k = 1, 2..., m; j = m+1, ..., n$ to $a_{kj} + \Delta$.

1. The feasibility of the solution and simplex multipliers remain unaffected.
2. The updated basis is $B(\Delta) = B + \Delta e_k e_i^T = B\left(I + \Delta B^{-1} e_k e_i^T\right)$, and $B^{-1}(\Delta) = \left(I - \varphi B^{-1} e_k e_i^T\right) B^{-1}$, where $\varphi = \left[\beta_{ik} + \Delta^{-1}\right]^{-1}$.
3. The updated solution is $x_B(\Delta) = x_B - \varphi x_i^* B_{\cdot k}^{-1}$, and the condition for primal feasibility is $\max_{\{q \in B | x_i^* \beta_{qk} < 0\}} \frac{x_q^*}{x_i^* \beta_{qk}} \leq \varphi \leq \min_{\{q \in B | x_i^* \beta_{qk} > 0\}} \frac{x_q^*}{x_i^* \beta_{qk}}$.
4. The simplex multipliers is $\lambda(\Delta) = \lambda - \varphi \lambda_k e_i^T B^{-1}$, and the reduced cost is $r_N(\Delta) = r_N - \varphi \lambda_i \left(B^{-1} N\right)_k$. And the condition for dual feasibility is $\max_{\{j \in N | \lambda_i a_{kj} < 0\}} \frac{r_j}{\lambda_i a_{kj}} \leq \varphi \leq \min_{\{j \in N | \lambda_i a_{kj} > 0\}} \frac{r_j}{\lambda_i a_{kj}}$.

4.4 Lagrange Duality

Lagrange dual problem is always a convex optimization problem regarding the dual variable, i.e., $\min_x \left(f(x) - \lambda^\top g(x)\right)$ is concave regarding λ. On the basis of weak duality theorem, we can derive the lower bound of the primal problem.

$$
\begin{array}{lll|ll}
 & \textbf{Primal} & & \textbf{Dual} & \\
\min & f(x) & & \max\limits_{\lambda \geq 0} & \left(\min\limits_x f(x) - \lambda^T g(x)\right) \\
\text{s.t.} & g_i(x) \geq 0, x \geq 0 & & \text{where} & g(x) = (g_1(x), ..., g_n(x))^T
\end{array}
\tag{4.15}
$$

Note on Sign of Lagrange multiplier *Be careful to the sign of Lagrange multiplier, since $g(x) \geq 0$ and $\lambda \geq 0$, and we want to minimize the objective function. According to the logic of penalty method, the objective function should minus $\lambda g(x)$ to ensure that $g(x) \geq 0$ holds when we minimize the objective function. Thus, if $g(x) \leq 0$, the Lagrange function should be $f(x) + \lambda g(x)$.*

Definition 4.9 (Lagrangian function for standard LP)

Suppose the original problem is the following, call this problem (P),

$$\text{minimize } f(x) \text{ subject to } h(x) = b, x \in X \quad L(x, \lambda) = f(x) + \lambda^T (h(x) - b)$$

then the Lagrangian of (P) is defined as

$$L(x, \lambda) = f(x) - \lambda^T (h(x) - b)$$

for $\lambda \in \mathbb{R}^m$. λ is known as the Lagrange multiplier.

Remark Note that the sign of λ does not change our result because of the equality constraint.

Theorem 4.6 (Lagrangian sufficiency)

Let $x^* \in X$ and $\lambda^* \in \mathbb{R}^m$ be such that

$$L\left(x^*, \lambda^*\right) = \inf_{x \in X} L\left(x, \lambda^*\right) \quad \textit{and} \quad h\left(x^*\right) = b$$

Then x^* is optimal for (P).

Proof ■

4.5 KKT condition

Chapter 5 Network Flow Optimization

5.1 Basic Algorithm and Theorem in Network

5.1.1 Complexity Analysis

5.1.2 Searching Algorithm

Search algorithm is to identify all nodes that can be reached by direct path. It is easy to see the search algorithm runs in $O(m+n)=O(m)$ time.

```
Begin
    Unmark all nodes in N;
    Mark node s; pred(s) = 0;
    next:=1; LIST:={s};
    While LIST≠ ϕ doo
    Begin
        Select a node i from LIST;
        If node i is incident to an admission arc (i,j), then
        Begin
            Mark node j;
            pred(j)=i;
            next:=next+1
            order(j):=next
            add j to LIST
        End
        Else delete node i from LIST
    End
End
```

5.1.3 Breath-first Search

Maintain the LIST as a queue, select nodes from the front of LIST and add them to the rear.

Lemma 5.1 (Breadth First Search theorem)

The breadth first search tree is the "shortest path tree", that is, the path from s to j in the tree has the fewest possible number of arcs.

Note on *Note that the shortest path tree here does not clarify the distance, i.e., each arc's distance is 1.*

5.1.4 Depth-first Search

Maintain the LIST as a stack, select nodes from the front of LIST and add them to the front. A depth-first order also satisfies the following properties,

- If node j is a descendant of node $i \neq j$, then order(j)>order(i).
- All the descendants of any node are ordered consecutively.

5.1.5 Acyclic Identification

Definition 5.1 (Topological Ordering)

The labeling of a graph is a topological ordering if every arc joins a lower-labeled node to a higher-labeled node. That is, for every $(i, j) \in A$, order(i)<order(j).

Proposition 5.1 (Unique Topological Ordering)

x

Proposition 5.2 (Topological ordering and acyclic)

A network is acyclic iff it possesses a topological ordering of its nodes.

Below is the *Topological sorting* algorithm to identify if the network is acyclic and give a topological ordering.

```
Begin
    For all i ∈ N, do indegree(i)=0;
    For all (i,j) ∈ A, do indegree(j)+=1;
    LIST:=ϕ; next:=0;
    For all i ∈ N do
    If indegree(i)=0, then LIST=LIST∪ {i};
    While LIST≠ ϕ do
    Begin
        Select a node i from LIST and delete it;
        next:=next+1; order(i) :=next;
        For (i,j) ∈ A, do
        Begin
            indegree(j)-=1;
            If indegree(i)=0, then LIST=LIST∪ {j};
        End
    End
    If next<n, then the network contains a cycle;
    Else it is acyclic, and the labeling is a topological ordering.
End
```

Proposition 5.3 (Adjacency matrix and acyclic)

A directed graph G is acyclic iff we can renumber its nodes so that its node-node adjacency matrix is a lower triangular matrix.

5.1.6 Flow Decomposition

Lemma 5.2 (Flow decomposition theorem 1)

Let $f \geq 0$ be a nonzero circulation. Then, there exist simple circulations $f^1, \ldots, f^k$, involving only forward arcs, and positive scalars $a_1, \ldots, a_k$, such that

$$f = \sum_{i=1}^{k} a_i f^i$$

Furthermore, if f is an integer vector, then each a_i can be chosen to be an integer.

Lemma 5.3 (Flow decomposition theorem 2)

Every path and cycle flow has a unique representation of non-negative arc flows. Let $f(p), p \in P$ and $f(w), w \in W$ be the path and cycle flows. Let $\delta_{ij}(p) = 1$ if $(i,j) \in p$ and 0 otherwise, $\delta_{ij}(w) = 1$ if $(i,j) \in w$ and 0 otherwise. Then,

$$x_{ij} = \sum_{p \in P} \delta_{ij}(p) f(p) + \sum_{w \in W} \delta_{ij}(w) f(w)$$

Conversely, every non-negative arc flow can be represented as a path and cycle flow (though not necessarily unique) with the following two properties:

- *Every directed path with positive flow connects a supply node to an excess node.*
- *At most n+m paths and cycles have non-zero flow. Out of these, at most m cycles have non-zero flow.*

Proof Note that each iteration we can construct a loop or a path to eliminate a node or an arc. And there are $n + m$ nodes and arcs, thus, it needs at most $n + m$ non-zero loop or path for iteration. And each time we construct a non-zero loop, we can remove an arc, thus, there are at most m non-zero loop we can construct. ■

There is also an algorithm to do *flow decomposition*.

```
x
```

5.2 Minimum Cost Flow Problem

In this problem, $b(i) > 0$ is a supply node, $b(i) < 0$ is a demand node. If a flow satisfies these constraints, it will be called a *feasible flow*.

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} \\
\text{s.t.} \quad & \sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = b(i), i \in N \quad \text{(Flow balance constraint)} \\
& l_{ij} \le x_{ij} \le u_{ij}, (i,j) \in A \quad \text{(Capacity constraint)}
\end{aligned} \tag{5.1}$$

By summing the *Flow balance constraint*, we obtain the assumption:

$$\sum_{i\in N} b(i) = 0$$

We also have a matrix form like this, where N is the node-arc incidence matrix.

$$\begin{aligned}
\min \quad & cx \\
\text{s.t.} \quad & Nx = b \quad \text{(Flow balance constraint)} \\
& l \le x \le u \quad \text{(Capacity constraint)}
\end{aligned} \tag{5.2}$$

Note that follow problems are special variants:

- Shortest path problem,
  - If we only want the solution from node s to node t, then set $b(s) = 1, b(t) = -1$ and $b(i) = 0$.
  - If we want all shortest path to node i, then set $b(s) = n - 1$ and $b(i) = -1 \forall i \neq s$.
- Maximum flow problem (Min cut), here we set $b(i) = 0 \forall i \in N$ and $c_{ij} = 0 \forall (i,j) \in A$, and introduce an additional arc (t,s) with cost $c_{ts} = -1$ and flow bound $u_{ts} = \infty$. Since any flow on arc (t,s) must travel from node s to node t through the arcs in A (since each b(i)=0), the minimum cost flow solution maximizes the flow on arc (t,s).
- Assignment problem, a special class of transportaion problem, here $x_{ij} = 0$ or 1.
- Transportation problem
- Circulation problem, here $b(i) = 0 \forall i \in N$ and we wish to find the circulation with minimum cost.
- Convex cost flow problems, here the cost is a convex function of the amount of flow.
- Generalized flow problems, here arcs may "consume" or "generate" flow, and arcs only conserve flows in the minimum cost flow problem. When x_{ij} units of flow enter an arc (i,j), then $\mu_{ij}x_{ij}$ units arrive at node j, we say the arc is lossy if $0 < \mu_{ij} < 1$ and gainy if $1 < \mu_{ij} < \infty$.
- Multicommodity flow problems

Note that every variant of the network flow problem can be shown to be equivalent to each other:

- Every network flow problem can be reduced to one with exactly one source and exactly one sink node.

- Every network flow problem can be reduced to one without sources or sinks, that is, we can transform the former to a circulation problem.
- Transformation of a node capacity into an arc capacity, just split this node into two nodes with an arc capacity equal to the node capacity.
- The lower bound of arc flow constraint can be reduced to zero, just construct the connection of $\mathbf{y}_{\mathrm{ij}} = \mathbf{x}_{\mathrm{ij}} - \boldsymbol{l}_{\mathrm{ij}}$.
- Inequality constraints $\sum_{\mathrm{j}} \mathbf{x}_{\mathrm{ij}} - \sum_{\mathbf{k}} \mathbf{x}_{\mathbf{ki}} \leq \mathbf{b}_{\mathbf{i}}$: Construct a "dummy node" n+1 and $\mathbf{b}_{n+1} = -B$, where $B = \sum_i b_i$. Any feasible solution for the original problem can be transformed into a feasible solution for the new problem by sending excess flow to node n+1.
- Eliminating upper bounds (Orlin, 2010, Lec. 4): For i with $b(i)$ and j with b(j) and arc with x_{ij}, we transform i with $b(i) - u_{ij}$ and j with $b(j)$ and a new node k with $u(ij)$ and $u_{ij} - x_{ij}$ from k to i and x_{ij} from k to j.

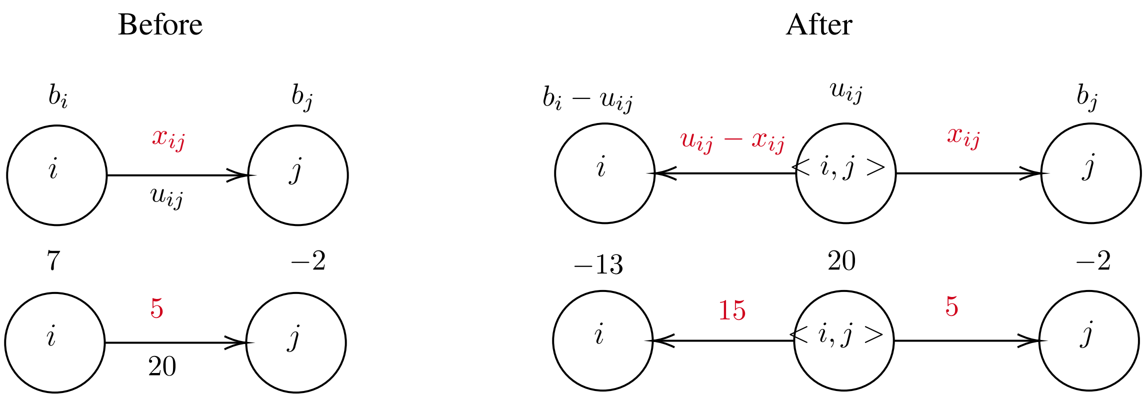

Figure 5.1: Eliminating upper bounds

- Undirected arcs to directed arcs, this is actually similar to the absolute case in LP. Suppose the arc {i,j} is undirected with cost $c_{ij} \geq 0$ and capacity u_{ij}, we replace each undirected arc by two directed arcs (i,j) and (j,i), both with cost c_{ij} and capacity u_{ij}.
- Arc Reversal (Ahuja et al., 1993, P. 40), this is typically used to remove arcs with negative costs. In this transformation we replace the variable x_{ij} by $u_{ij} - x_{ji}$. Doing so replaces the arc (i,j), which has an associated cost c_{ij}, by the arc (j,i) with an associated cost $-c_{ij}$.

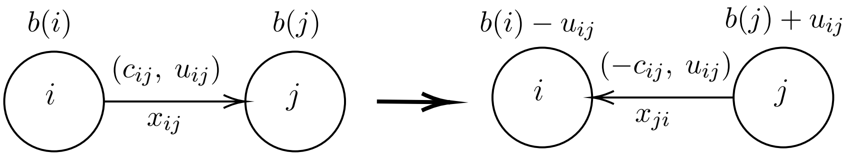

Figure 5.2: Arc reversal transformation.

There are also two kinds network models does not correlate to flow problems.

- Minimum spanning tree problem
- Matching problem

Definition 5.2 (Circulation)

Any flow vector $\boldsymbol{f}$ *that satisfies* $Af = 0$ *is called a circulation.*

Intuitively, with zero external supply and demand, the flow "circulates" inside the network.

5.3 Shortest Path Problem

There is some assumptions for this problem,

- All arc lengths are integers. (Can be relaxed)
- The network contains a directed path from node s to every other node in the network.
- The network does not contain a negative cycle.
- The network is directed.

Here we use d(i) denotes the length of some path from s node to node i. And the procedure update(i) means that if $d(j) > d(i) + c_{ij}$ then do $d(j) := d(i) + c_{ij}$ and pred(j):=i, note that distance labels can only decrease in an update step.

Proposition 5.4 (Optimality for subpath)

If the path $s = i_1 - ... - i_k = k$ *is a shortest path from node s to node k, then for every* $q = 2, ..., k-1$*, the subpath* $s = i_1 - ... - i_q$ *is the shortest path from node s to node* i_q*.*

Proposition 5.5 (Optimality condition 1)

A direct path P *from the source node to node* k *is a shortest path iff* $d(j) = d(i) + c_{ij}, \forall (i,j) \in P$*, here d(.) denotes the shortest path distance.*

Proof If side: Sum up equations $\forall (i,j) \in P$, then you have $d(k) = c_{12} + ...c_{k-1,k}$, and d(.) denotes the shortest path distance, this means that this path is a shortest path.

Onlyif side: ■

Proposition 5.6 (Optimality condition 2 (Malik et al., 1989))

Consider a network without any negative cost cycle. For every node $j \in N$*, let* $d^s(j)$ *denote the length of a shortest path from node s to node j and let* $d^t(j)$ *denote the length of a shortest path from node j to node t.*

- *An arc (i,j) is on a shortest path from node s to node t iff* $d^s(t) = d^s(i) + c_{ij} + d^t(j)$*.*
- $d^s(t) = min\{d^s(i) + c_{ij} + d^t(j), (i,j) \in A\}$*.*

There are two kinds of algorithms for solving shortest path problems: label setting and label correcting. The approaches vary in how they update the distance labels from step to step and how they "converge" toward the shortest path distances. Label-setting algorithms designate one label as permanent (optimal) at each iteration. In contrast, label-correcting algorithms consider all labels as temporary until the final step. Label-setting can only apply to acyclic networks and problems with nonnegative arc lengths, while label-correcting are more general and apply to all classes of problems.

5.3.1 Floyd-Warshall algorithm

Here we use *Floyd-Warshall algorithm* to derive the shortest path, and this is also applicable to the networks with negative arcs

Set $d(s) = 0$ and the remaining distance labels to very large numbers.
Examine the nodes in the topological order and for each order i, scan the arcs in A(i).
If $d(j) > d(i) + c_{ij}$, for any $(i, j) \in A(i)$, then update $d(j) = d(i) + c_{ij}$.
After examining all nodes, the distance labels is optimal.

5.3.2 Dijkastra's Algorithm

```
Begin
    S := φ; S̄ := N;
    d(i) := ∞, ∀i ∈ N;
    d(s) := 0, pred(s):=0;
    While |S|<n, do
    Begin
        Let node i ∈ S̄ be such that d(i) = Min{d(j) : j ∈ S̄}
        S := S ∪ {i}; S̄ := S̄ − {i};
        For each (i, j) ∈ A(i), do
            If d(j) > d(i) + c_ij, then d(j) = d(i) + c_ij,
            pred(j):=i;
    End
End
```

Proof [(Borradaile, n.d.)] Reference. ■

This algorithm, also known as label-setting algorithm, maintains two sets of nodes: permanently labeled nodes S and temporarily labeled nodes $\bar{S}$ at each iteration. And the most time-consuming step is at node selection due to distance-label comparison.

Proposition 5.7

The distance labels that the Dijkstra's algorithm designates as permanent are non-decreasing.

Proposition 5.8

If d(i) is the distance label that the algorithm designates as permanent at the beginning of an iteration, then at the end of the iteration, $d(j) \leq d(i) + C$ for each finitely labeled node $j \in \bar{S}$, where C is the maximum arc length.

5.3.3 Improved Dijkastra's Algorithm

Here we propose some data structures to improve Dijkastra's Algorithm's efficiency. One way is using *Buckets* in Dial's Algorithm.

5.3.4 Label Correcting Algorithm

Correcting Algorithm is more complicated and can be applied to more general case.

Theorem 5.1 (Optimality Condition)

For every node $j \in N$, let $d(j)$ denote the length of some directed path from the source node to node j. Then, $d(j)$ represents the shortest path distances iff they satisfy the following optimality condition:

$$d(j) \leq d(i) + c_{ij}, (i, j) \in A$$

Actually, this condition can be interpreted as reduced cost condition,, we can define the reduced cost length c_{ij}^d of arc (i, j), where $c_{ij}^d = c_{ij} + d(i) - d(j)$.

Lemma 5.4 (reduced cost property)

- *For any directed cycle W, $\sum_{(i,j)\in W} c_{ij}^d = \sum_{(i,j)\in W} c_{ij}$.*
- *For any directed path P from node k to node l, $\sum_{(i,j)\in P} c_{ij}^d = \sum_{(i,j)\in P} c_{ij} + d(k) - d(l)$.*
- *If $d(.)$ represent shortest path distances, $c_{ij}^d \geq 0$ for every arc $(i, j) \in A$.*

Below is the generic algorithm

```
Begin
    d(s) := 0; pred(s) := 0;
    d(i) := ∞, for i ∈ N − {s};
    While some arc (i,j) satisfies d(j) > d(i) + c_ij, do
    Begin
        d(j) = d(i) + c_ij, pred(j) := i;
    End
End
```

Note on

- *The predecessor indices might not necessarily define a tree. In case of a negative cycle, the resulting list can form a disconnected graph.*
- *We refer to the collection of arcs (pred(j),j) as the predecessor graph, and the label-correcting algorithm satisfies the invariant property that for every arc (i, j) in the predecessor graph, $c_{ij}^d \leq 0$. When the algorithm terminates, the reduced arc length in the predecessor tree must be zero.*

Below is a modified label-correcting algorithm, since the generic algorithm does not specify any method for selecting an arc violating the optimality condition.

```
Begin
    d(s) := 0; pred(s) := 0;
    d(i) := ∞, for i ∈ N − {s};
    LIST:= {S};
    While LIST≠ ∅, do
    Begin
        Remove an element i from LIST
        For each arc (i,j) ∈ A(i), do
            If d(j) > d(i) + c_ij, then
            Begin
                d(j) = d(i) + c_ij; pred(j) := i;
                If j ∉ LIST, then add j to LIST
            End
    End
End
```

5.3.5 Connection to other topic

5.3.5.1 Dynamic Lot Sizing

5.3.5.2 Most vital arc problem (Malik et al., 1989)

5.3.5.3 Kth shortest path problem

Note that even there are many shortest paths, this algorithm works.

5.4 Maximum Flow Problem

$$
\begin{aligned}
\max \quad & v \\
\text{s.t.} \quad & \sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = \begin{cases} v & \text{for } i = s \\ 0 & \text{for all } i \in N - \{s \text{ and } t\} \\ -v & \text{for } i = t \end{cases} \qquad (5.3) \\
& 0 \le x_{ij} \le u_{ij} \quad \text{for each } (i,j) \in A
\end{aligned}
$$

Assumption 5.1

- *The network is directed. (feasibility)*
- *All the capacities are non-negative integers. (feasibility)*
- *The network does not contain a directed path from node s to node t consisting of*

infinite capacity. (bounded, finite optimal)

- *The network does not contain parallel arcs.*

Definition 5.3 (Residual Capacity and Residual Network)

Given a flow x, the residual capacity $r_{ij} = u_{ij} - x_{ij} + x_{ji}$ of arc $(i, j) \in A$ is the maximum additional flow that can be sent from the arcs (i, j) and (j, i) between nodes i and j. Here r_{ij} has two components

- *$u_{ij} - x_{ij}$ is the unused capacity of (i, j).*
- *the current flow x_{ji} on arc (j, i), which can cancel the increase in the flow from i to j.*

We refer to the network $G(x)$ consisting of the arcs with positive residual capacities as the residual network.

By definition, we have $x_{ij} - x_{ji} = u_{ij} - r_{ij}$, since x_{ij} and x_{ji} are positive here, if $u_{ij} \geq r_{ij}$, $x_{ij} = u_{ij} - r_{ij}$ and $x_{ji} = 0$, if $u_{ij} < r_{ij}$, $x_{ji} = r_{ij} - u_{ij}$ and $x_{ij} = 0$.

Definition 5.4 (s-t Cut)

A cut is an $s-t$ cut if $s \in S$ and $t \in \bar{S}$. Capacity of an $s-t$ cut $u[S, \bar{S}] = \sum_{(i,j)\in(S,\bar{S})} u_{ij}$, and this is the upper bound of the flow from s to t. Residual capacity of an $s-t$ cut is $r[S, \bar{S}] = \sum_{(i,j)\in(S,\bar{S})} r_{ij}$.

Let x be a flow in the network. the amount of flow from nodes in S to nodes in $\bar{S}$ can be expressed as follows. Since $0 \leq x_{ij} \leq u_{ij}$, we have $v \leq U[S, \bar{S}]$.

$$v = \sum_{i\in S}\left[\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji}\right] = \sum_{(i,j)\in(S,\bar{S})} x_{ij} - \sum_{(i,j)\in(\bar{S},S)} x_{ij}$$

Lemma 5.5 (Cut's Property)

- *The value of any flow is less than or equal to the capacity of any cut in the network.*
- *For any flow x of value v in a network, the additional flow that can be sent from the source node s to the sink node t is less than or equal to the residual capacity of any $s-t$ cut.*

Any flow x whose value equals the capacity of some cut $[S, \bar{S}]$ is the maximum flow and the cut is the minimum cut. That is, the minimum cut problem is the dual problem of maximum flow problem.

Below is the Generic Augmenting Path Algorithm, Labeling Algorithm and Procedure Augment.

```
Begin
    x := 0;
    while G(x) contains a path from s to t, do
```

```
    Begin
        Identify an augmenting path P from s to t
        δ := Min {r_ij : (i,j) ∈ P}.
        Augment δ units of flow along P and update G(x)
    End
End

Begin
    Label node t;
    While t is labeled, do
    Begin
        Unlabel all the nodes;
        Set pred(j) := 0 for j ∈ N
        Label node s, and LIST:= {s};
        While LIST≠ ∅ and t is unlabeled, do
        Begin
            Remove a node i from LIST;
            For each arc (i,j) in the residual network; do
                If j is unlabeled, set pred(j) := i, label j, add j to LIST
        End
        If t is labeled, then augment.
    End
End

Begin
    Use predecessor labels to trace back from the sink to the source to obtain a
        path P;
    δ := Min {r_ij : (i,j) ∈ P}
    Augment along P
End
```

5.4.1 Dual: Min-Cut

The dual can be formulated as this way[1] or this way[2], this way[3], this way[4].

A second explanation of Dual[5].

[1]Lecture 15, Stanford University — CS261: Optimization

[2]The dual of the maximum flow problem

[3]Lecture 24: The Max-Flow Min-Cut Theorem Math 482: Linear Programming

[4]Lecture 14: Linear Programming II

[5]Lecture 10: Duality in Linear Programs

Theorem 5.2 (Max-Flow Min-Cut Theorem)

The maximum value of the flow from a source node s to a sink node t in a capacitated network equals the minimum capacity among all $s-t$ cuts.

Theorem 5.3 (Augmenting Path Theorem)

A flow x^* is a maximum flow iff the residual network $G(x^*)$ contains no augmenting path.

Theorem 5.4 (Integrality Theorem)

If all arc capacities are integer, the maximum flow problem has an integer maximum flow.

5.5 Maximum Flow Problem

$$\begin{aligned}
\max \quad & v \\
\text{s.t.} \quad & \sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = \begin{cases} v & \text{for } i = s \\ 0 & \text{for all } i \in N - \{s \text{ and } t\} \\ -v & \text{for } i = t \end{cases} \\
& 0 \le x_{ij} \le u_{ij} \quad \text{for each } (i,j) \in A
\end{aligned} \tag{5.4}$$

Assumption 5.2

- *The network is directed. (feasibility)*
- *All the capacities are non-negative integers. (feasibility)*
- *The network does not contain a directed path from node s to node t consisting of infinite capacity. (bounded, finite optimal)*
- *The network does not contain parallel arcs.*

Definition 5.5 (Residual Capacity and Residual Network)

Given a flow x, the residual capacity $r_{ij} = u_{ij} - x_{ij} + x_{ji}$ of arc $(i,j) \in A$ is the maximum additional flow that can be sent from the arcs (i,j) and (j,i) between nodes i and j. Here r_{ij} has two components

- *$u_{ij} - x_{ij}$ is the unused capacity of (i,j).*
- *the current flow x_{ji} on arc (j,i), which can cancel the increase in the flow from i to j.*

We refer to the network $G(x)$ consisting of the arcs with positive residual capacities as the residual network.

By definition, we have $x_{ij} - x_{ji} = u_{ij} - r_{ij}$, since x_{ij} and x_{ji} are positive here, if $u_{ij} \ge r_{ij}$, $x_{ij} = u_{ij} - r_{ij}$ and $x_{ji} = 0$, if $u_{ij} < r_{ij}$, $x_{ji} = r_{ij} - u_{ij}$ and $x_{ij} = 0$.

Definition 5.6 (s-t Cut)

A cut is an $s-t$ cut if $s \in S$ and $t \in \bar{S}$. Capacity of an $s-t$ cut $u[S,\bar{S}] = \sum_{(i,j)\in(S,\bar{S})} u_{ij}$, and this is the upper bound of the flow from s to t. Residual capacity of an $s-t$ cut is $r[S,\bar{S}] = \sum_{(i,j)\in(S,\bar{S})} r_{ij}$.

Let x be a flow in the network. the amount of flow from nodes in S to nodes in $\bar{S}$ can be expressed as follows. Since $0 \le x_{ij} \le u_{ij}$, we have $v \le U[S,\bar{S}]$.

$$v = \sum_{i\in S}\left[\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji}\right] = \sum_{(i,j)\in(S,\bar{S})} x_{ij} - \sum_{(i,j)\in(\bar{S},S)} x_{ij}$$

Lemma 5.6 (Cut's Property)

- *The value of any flow is less than or equal to the capacity of any cut in the network.*
- *For any flow x of value v in a network, the additional flow that can be sent from the source node s to the sink node t is less than or equal to the residual capacity of any $s-t$ cut.*

Any flow x whose value equals the capacity of some cut $[S,\bar{S}]$ is the maximum flow and the cut is the minimum cut. That is, the minimum cut problem is the dual problem of maximum flow problem.

Below is the Generic Augmenting Path Algorithm, Labeling Algorithm and Procedure Augment.

```
Begin
    x := 0;
    while G(x) contains a path from s to t, do
    Begin
        Identify an augmenting path P from s to t
        δ := Min {r_ij : (i,j) ∈ P}.
        Augment δ units of flow along P and update G(x)
    End
End

Begin
    Label node t;
    While t is labeled, do
    Begin
        Unlabel all the nodes;
        Set pred(j) := 0 for j ∈ N
        Label node s, and LIST:= {s};
        While LIST≠ ∅ and t is unlabeled, do
        Begin
            Remove a node i from LIST;
```

```
        For each arc (i,j) in the residual network; do
            If j is unlabeled, set pred(j) := i, label j, add j to LIST
        End
        If t is labeled, then augment.
    End
End

Begin
    Use predecessor labels to trace back from the sink to the source to obtain a
         path P;
    δ := Min {r_ij : (i,j) ∈ P}
    Augment along P
End
```

5.5.1 Dual: Min-Cut

The dual can be formulated as this way[6] or this way[7], this way[8], this way[9].

A second explanation of Dual[10].

Theorem 5.5 (Max-Flow Min-Cut Theorem)

The maximum value of the flow from a source node s to a sink node t in a capacitated network equals the minimum capacity among all $s-t$ cuts.

Theorem 5.6 (Augmenting Path Theorem)

A flow x^* is a maximum flow iff the residual network $G(x^*)$ contains no augmenting path.

Theorem 5.7 (Integrality Theorem)

If all arc capacities are integer, the maximum flow problem has an integer maximum flow.

5.6 Network Simplex Algorithm

Definition 5.7 (Free arc and restricted arc)

Arc (i,j) is free if $0 < x_{ij} < u_{ij}$ and is a restricted arc if $x_{ij} = 0$ or $x_{ij} = u_{ij}$.

Definition 5.8 (Cycle-free solution)

A solution x is cycle-free if the network contains no cycle composed only of free arcs.

[6] Lecture 15, Stanford University — CS261: Optimization

[7] The dual of the maximum flow problem

[8] Lecture 24: The Max-Flow Min-Cut Theorem Math 482: Linear Programming

[9] Lecture 14: Linear Programming II

[10] Lecture 10: Duality in Linear Programs

Definition 5.9 (Spanning tree solution)

A feasible solution x and the associated spanning tree of the network is a spanning tree solution if every non-tree arc is a restricted tree. A spanning tree solution partitions the arc set A into three sets (T, L, U):

- *T: $n-1$ arcs in the spanning tree.*
- *L: the non-tree arcs whose flows are restricted to be zero.*
- *U: the non-tree arcs whose flows are restricted to be the arcs' flow capacities.*

A spanning tree structure is feasible if all arcs' flow satisfy the bounds. The spanning tree is non-degenerate if every tree arc in a spanning tree solution is a free arc.

Lemma 5.7 (Cycle Free Property)

If the objective function of a minimum cost flow problem is bounded from below over the feasible region, the problem always has an optimal cycle free solution.

Lemma 5.8 (Spanning Tree Property)

If the objective function of a minimum cost flow problem is bounded from below over the feasible region, the problem always has an optimal spanning tree solution.

Note on *Similar to simplex method of LP, we can construct a spanning tree solution as a basic solution, e.g., we can set $x_{ij} = 0$ for $(i,j) \in L$, $x_{ij} = u_{ij}$ for $(i,j) \in U$ and solve x_{ij} for $(i,j) \in T$.*

Theorem 5.8 (Optimality Condition)

A spanning tree structure (T, L, U) is an optimal spanning tree structure of the minimum cost flow problem if it is feasible and for some choie of node potential π, the arc reduced costs c^{π}_{ij} satisfy the following conditions:

- $c^{\pi}_{ij} = 0$ *for all* $(i,j) \in \mathbf{T}$.
- $c^{\pi}_{ij} \geq 0$ *for all* $(i,j) \in \mathbf{L}$.
- $c^{\pi}_{ij} \leq 0$ *for all* $(i,j) \in \mathbf{U}$.

Below is the procedure for computing node potentials, where thread(i) is the node in the depth-first traversal search encountered after the node itself.

```
Begin
    π(1) = 0;
    j = thread(1);
    While j ≠ 1, do
        Begin
            i := pred(j);
            If (i,j) ∈ A, then π(j) := π(i) − c_ij;
            If (j,i) ∈ A, then π(j) := π(i) + c_ij;
            j = thread(j).
```

```
        End
End
```

Lemma 5.9 (Dual Integrality Property)

If all arc costs are integer, the minimum cost flow problem always has optimal integer node potentials.

Lemma 5.10 (Primal Integrality Property)

Below is the procedure for computing flows and the Network Simplex Algorithm.

```
Begin
    b^{\prime}(i)=b(i), i \in N;
    For (i,j) ∈ U, do
        set x_ij = u_ij, b'(i) = b'(i) − u_ij, b'(j) = b'(j) + u_ij;
    For (i,j) ∈ L, do
        set x_ij = 0;
    T' := T;
    While T' ≠ {1} do
        Begin
            Select a leaf node j ∈ T';
            i := pred(j);
            If (i,j) ∈ T', then
                x_ij := −b(j);
            Else
                x_ij := b(j).
            b'(i) := b'(i) + b'(j);
            Delete node j and the arc incident to it in T'.
        End
End


Begin
    Determine an initial feasible tree structure (T, L, U);
    Let x be the flow and π the node potentials associated with tree;
    While some non-tree arcs violate optimality condition, do
        Begin
            Select an entering arc (k,l) violating the optimality condition;
            Add (k,l) to the tree and determine the leaving (p,q);
            Perform a tree update, update the flow x and node potential π.
        End
End
```

Note on Entering variable *Choosing* $(i,j) \in L$, *with* $c_{ij}^{\pi} < 0$ *or* $(i,j) \in U$, *with* $c_{ij}^{\pi} > 0$. *The* *standard for selecting can be either the largest* $\left|c_{ij}^{\pi}\right|$ *or the first arc scanned.*

Note on Pivoting *Suppose we choose* (k,l) *as entering variable, and after that we get the cycle* w*, which is also called as pivot cycle.*

- *Let the orientation of the cycle* W *be that of* (k,l) *if* $(k,l) \in L$ *or the opposite to that of* (k,l) *if* $(k,l) \in U$.
- $\bar{W}$ *and* $\underline{W}$ *are respectively the forward and backward arc sets.*
- *The maximum flow change* δ_{ij} *satisfies that* $\delta_{ij} = u_{ij} - x_{ij}$ *if* $(i,j) \in \bar{W}$*, and* $\delta_{ij} = x_{ij}$ *if* $(i,j) \in \underline{W}$.
- *Augment* $\delta = \text{Min}\,\{\delta_{ij} : (i,j) \in W\}$*, and the arc that defines* δ *leaves the basis.*

5.7 Lagrangian Relaxation

If LP's constraints can be divided into two types: some are easy to solve, and the others are not easy to solve, than we can use Lagrangian relaxation to remove "bad" constraints and putting them into the objective function, assigned with weights (the Lagrangian multiplier).

5.7.1 Symmetric Form

	Primal		Lagrangian Relaxation	Lagrangian multiplier problem
min	$c^T x$	min	$cx + \mu(Ax - b)$	$L^* = \max\limits_{\mu} L(\mu)$
s.t.	$Ax = b$	s.t.	$x \in X$	$L(\mu) = \min\{cx + \mu(Ax - b) : x \in X\}$
	$x \in X$ a polyhedral set.			

(5.5)

Theorem 5.9 (Lagrangian Bounding Pricinple)

For any vector μ *of the Lagrangian multipliers, the value* $L(\mu)$ *of the Lagrangian function is a lower bound on the optimal objective function value* z^* *of the original optimization problem.*

Theorem 5.10 (Weak Duality)

The optimal objective function value L^* *of the Lagrangian multiplier problem is always a lower bound on the optimal objective function value of the original problem (i.e.,* $L^* \leq z^*$*).*

Theorem 5.11 (Optimality Test)

- *Suppose that* μ *is a vecotr of Lagrangian multipliers and* x *is a feasible solution to the Primal problem satisfying the condition* $L(\mu) = cx$*. Then* $L(\mu)$ *is an optimal solution of the Lagrangian multiplier problem (i.e.* $L^* = L(\mu)$*) and* x *is an optimal solution to the Primal problem.*
- *If for some choice of the Lagrangian multiplier vector* μ*, the solution* x^* *of the*

Lagrangian relaxation is feasible in the Primal problem, then x^* is an optimal solution to the Primal problem and μ is an optimal solution to the Lagrangian multiplier problem.

5.7.2 Asymmetric Form

$$
\begin{array}{llcll}
 & \textbf{Primal} & & & \textbf{Lagrangian Dual} \\
\min & z(x) = c^T x & & \max & f(w) = w^T b + \min_{x \in X}(c^T - w^T A)x \\
\text{s.t.} & Ax \geq b & & \text{s.t.} & w \geq 0 \\
 & x \in X, \text{ where } X \text{ is a polyhedral set.} & & &
\end{array} \tag{5.6}
$$

Theorem 5.12 (Weak Duality)

The optimal objective function value L^* of the Lagrangian multiplier problem is always a lower bound on the optimal objective function value of the original problem (i.e., $f(w^*) \leq z(x^*)$).

Proof This is equal to show any feasible solution x_0 to Primal and any feasible solution w_0 to Lagrangian Dual satisfy $c^T x_0 \geq f(w_0)$. Since x_0 is feasible to Primal, $x_0 \in X$ and $\min_{x \in X}(c^T - w^T A)x \leq (c^T - w^T A)x_0$. Note that $Ax_0 \geq b$ means $w_0^T A x_0 \geq w_0^T b \quad (w_0 \geq 0)$, thus

$$f(w_0) \leq w_0^T b + (c^T - w^T A)x_0 = c^T x_0 + w_0^T b - w^T A x_0 \leq c^T x_0$$

■

Theorem 5.13 (Strong Duality)

Suppose that X is nonempty and bounded and that the primal problem possess a finite optimal solution. Then

$$\min_{Ax \geq b, x \in X} c^T x = \max_{w \geq 0} f(w)$$

Proof

$$
\begin{array}{llll}
 & \textbf{Primal} & & \textbf{Dual} \\
\min & z(x) = c^T x & \max & \lambda_1^T b + \lambda_2^T d \\
\text{s.t.} & Ax \geq b & \text{s.t.} & \lambda_1^T A + \lambda_2^T B = c^T \\
 & Bx \geq d & & \lambda_1, \lambda_2 \geq 0
\end{array} \tag{5.7}
$$

Note that $x \in X$ can be expressed as $Bx \geq d$, and assume x^* is a feasible optimal solution to Primal, λ_1^* and λ_2^* are dual vector for constraint $Ax \geq b$ and $Bx \geq d$, then we must have dual feasiblity

$$(\lambda_1^*)^T A + (\lambda_2^*)^T B = c^T \tag{5.8}$$

and following complementary slackness conditions

$$\begin{cases} \lambda_1^*(Ax - b) = 0 \\ \lambda_2^*(Bx - d) = 0 \\ x^*((\lambda_1^*)^T A + (\lambda_2^*)^T B - c^T) = 0 \end{cases}$$

Since $\lambda_1^* \geq 0$, λ_1^* is also a feasible solution to $\max_{w \geq 0} f(w)$.

$$f(\lambda_1^*) = (\lambda_1^*)^T b + \min_{x \in X}(c^T - (\lambda_1^*)^T A)x$$

Consider the following duality, note that x^* and λ_2^* are optimal solution to Primal and Dual respectively because of primal feasiblity, dual feasibility (5.8) and complementary slackness conditions (5.7.2, 5.7.2).

$$\begin{array}{llll} & \textbf{Primal} & & \textbf{Dual} \\ \min & z(x) = (c^T - (\lambda_1^*)^T A)x & \max & \lambda_2^T d \\ \text{s.t.} & Bx \geq d & \text{s.t.} & \lambda_2^T B = c^T - (\lambda_1^*)^T A, \lambda_2 \geq 0 \end{array} \tag{5.9}$$

Thus $f(\lambda^*) = c^T x^*$, and by weak duality theorem we know $f(w) \leq c^T x^* = f(\lambda^*) = c^T x$, thus λ^* is also the optimal solution to the Lagrangian dual and the optimal solutions for both questions are equal. ∎

Chapter 6 Robust Optimization

Appendix Topology

A.1 Convex set, combination, hull and cone

Definition A.1 (Convex set)

A set $S \subset R^n$ is convex if $\forall \lambda \in [0,1]$ and $\forall x, x' \in S$, $\lambda x + (1-\lambda)x' \in S$.

Lemma A.1 (Convex set's property)

1. *$C :=$ convex set, $\beta :=$ real number, then $\beta C = \{x : x = \beta c, c \in C\}$ is convex.*
2. *$C, D :=$ convex set, then $C + D = \{x : x = c + d, c \in C, d \in D\}$ is convex.*
3. *$S, T :=$ convex set, then $S \cap T$ is a convex set.*

Lemma A.2

$C :=$ convex set, $y :=$ a point exterior to the closure of C. Then there is a vector a such that $a^T y < \inf_{x \in C} a^T x$.

Proof

$$a^T y < \inf_{x \in C} a^T x \iff \inf_{x \in C} (a^T x - a^T y) > 0 \iff \inf_{x \in C} a^T (x - y) > 0$$

That is equal to show that there exists $a^T(x-y) > 0$, i.e., the included angle is acute. Define $f(x) = ||x - y||$ (norm/distance). We want to $\min_{x \in C} f(x)$, find the point $x \in C$ closer to y. Since C is closure, there must be an optimal solution x^0, and $||x^0 - y|| \leq ||x - y|| \quad \forall x \in C$. Given x_0, let $x \in C$, then $\forall 0 < \alpha < 1$, $x_0 + \alpha(x - x_0) \in C$ (Convex set definition). And

$$||x_0 + \alpha(x - x_0) - y|| \geq ||x_0 - y||$$

Expanding the inequation then we have $\alpha||x - x_0||^2 + 2|x_0 - y|^T(x - x_0) \geq 0$, let $\alpha \to 0$, we have $|x_0 - y|^T(x - x_0) \geq 0$, that is

$$(x_0 - y)^\top x \geqslant (x_0 - y)^\top x_0 = (x_0 - y)^T (x_0 - y + y) = (x_0 - y)^\top (x_0 - y) + (x_0 - y)^\top y$$

Let $a = (x_0 - y)$, we have $a^\top x \geqslant a^T a + a^\top y$, since $a^T a$ is positive, a is what we want.

■

Definition A.2 (Convex combination)

$y = \sum_{i=1}^m \lambda_i y_i$ is a convex combination of $y_1, ..., y_m$ if $\lambda_i \geq 0$, $\sum_{i=1}^m \lambda_i = 1$.

Definition A.3 (Cone, Convex Cone)

1. *$C \subset R^n$ is a cone if $\forall x \in C, \alpha > 0$, $\alpha x \in C$.*
2. *$C \subseteq R^n$ is a convex cone if $\forall x, y \in C, \alpha, \beta \geq 0$, $\alpha x + \beta y \in C$.*

Example A.1

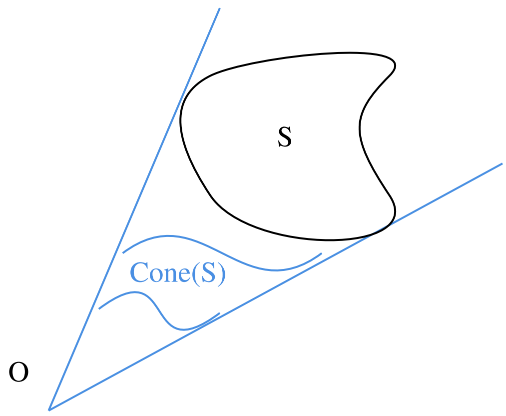

Figure A.1: Conic Hull

Definition A.4 (Convex hull)

Q is a convex hull of $v_1, .., v_k$ *if* $Q = \{v \in \Re^n : v \text{ is a convex combination of } v_1, v_2, \ldots, v_k\}$*, and we write* $Q = \operatorname{conv}(v_1, v_2, \ldots, v_k)$.

Note on *The convex hull of* $S \subseteq \mathbb{R}^n$ *is the smallest convex set containing* S.

Property

1. *Intersection of all convex sets containing* S.
2. *The set of all convex combinations of points in* S.

Theorem A.1 (Convex set and convex hull)

A set is convex iff convexhull(S)=S.

Definition A.5 (Conic Hull, Closure of Cone)

1. *Given a set* S*, the conic hull of* S*, denoted by* $cone(S)$*, is the set of all conic combinations of the points in* S*, i.e., the smallest convex cone included* S.
$$\operatorname{cone}(S) = \left\{ \sum_{i=1}^{n} \alpha_i x_i \mid \alpha_i \geq 0, x_i \in S \right\}$$
2. *Closure of* $cone(S)$:= *the closed convex hull of* S.

Note on *Conic hull is convex and includes the zero point.*

Lemma A.3

A closed bounded convex set in R^n *is equal to the closed convex hull of its extreme points.*

A.2 Hyperplane and Polytope

Theorem A.2 (Projection Lemma?)

Let $X \in R^m$ be a nonempty closed convex set, and let $y \notin X$. Then there exists $x^* \in X$ with minimum distance from y, moreover, for all $x \in X$ we have $(y - x^*)^{\mathrm{T}} (x - x^*) \leq 0$.

Definition A.6 (Hyperplane)

1. *A set $H \subset R^n$ is a hyperplane := $H = \{x \in R^n : \sum_{i=1}^n \alpha_i x_i = \beta\}$ for some $\beta \in R$ and some $(\alpha_1, \dots, \alpha_n) \subset R^n$ such that $\alpha_i \neq 0$ for some i.*
2. *Positive half space of H: $H^+ := \{x \in R^n : \sum_{i=1}^n \alpha_i x_i \geqslant \beta\}$.*
3. *Negative half space of H: $H^- := \{x \in R^n : \quad \sum_{i=1}^n \alpha_i x_i \leqslant \beta\}$*

Example A.2

1. If $n = 1$, H contains the point $\frac{\beta}{\alpha}$.
2. If $n = 2$ and $\alpha_1, \alpha_2 \neq 0$, H is the line $\alpha_1 x_1 + \alpha_2 x_2 = \beta$.

Property

1. $H^+ \cup H^- = R^n, \quad H^+ \cap H^- = H$
2. *A hyperplane H and its associated half spaces H^+ and H^- are convex sets.*

Lemma A.4

C := convex set, y := a boundary point of C. Then there is a hyperplane containing y and containing C in one of its closed half space.

Proof Let H starts as the sequence of $\{y_0, y_1, \dots, y\}$, according to lemma A.2, $\forall y_k$, we have $a_k^T y_k < \inf_{x \in C} a_k^T x$, and converge to y we have $a^T y < \inf_{x \in C} a^T x$, that is, $\forall y_k$, $a_k = x_{0k} - y_k$, and converge to for y, $a = 0$. The hyperplane $a^T y$ is what we want. ■

Theorem A.3 (Separating Hyperplane Theorem (Ali Ahmadi, 2016, Lec. 5))

1. *If S and T are two disjoint convex sets in R^n then there is a hyperplane $H \subset R^n$ such that $S \subset H^+$ and $T \subset H^-$.*
2. *Let C and D be two convex sets in R^n that do not intersect (i.e., $C \cap D = \emptyset$). Then, there exists $a \in R^n$, $a \neq 0$, $b \in R$, such that $a^T x \leq b$ for all $x \in C$ and $a^T x \geq b$ for all $x \in D$.*
3. *Special Case: Let C and D be two closed convex sets in R^n with at least one of them bounded, and assume $C \cap D = \emptyset$. Then $\exists a \in \mathbb{R}^n, a \neq 0, b \in \mathbb{R}$ such that*

$$a^T x > b, \forall x \in D \text{ and } a^T x < b, \forall x \in C$$

Note on Equality *Note that the equality in this theorem cannot be neglected (Ali Ahmadi, 2016, Lec. 5). For example, for $A = (x, y) : y \geq 0 \forall x \leq 0, y > 0 \forall x > 0$, then we can find $a = (1, 0)^T, b = 0$ to separate $A, \overline{A}$. However, there does not exists such a, b to separate withtout equality. The case of strict separate, i.e., $a^T x < b$ and $a^T x > b$ hold simultaneously, may not* *exist.*

Proof [Special Case (Ali Ahmadi, 2016, Lec. 5)] Let $c \in C$ and $d \in D$ be the points with the minimal distance, i.e.,

$$\begin{aligned} \operatorname{dist}(C, D) = \inf \ & \|u - v\| \\ \text{s.t. } & u \in C, v \in D \end{aligned}.$$

Furthermore, let

$$a = d - c, b = \frac{\|d\|^2 - \|c\|^2}{2}.$$

Then $f(x) = a^T x - b$ is the separating hyperplane what we want. We claims that

$$f(x) > 0, \forall x \in D \text{ and } f(x) < 0, \forall x \in C.$$

Note that we choose a to be perpendicular to dc, and b to ensure

$$f\left(\frac{c+d}{2}\right) = (d-c)^T \left(\frac{c+d}{2}\right) - \frac{\|d\|^2 - \|c\|^2}{2} = 0.$$

Then we can prove $f(x) > 0, \forall x \in D$ and $f(x) < 0, \forall x \in C$. Suppose for the sake of contradiction that $\exists \bar{d} \in D$ with $f(\bar{d}) \leq 0$, i.e., $(d-c)^T \bar{d} - \frac{\|d\|^2 - \|c\|^2}{2} \leq 0$. Since for $g(x) = \|x - c\|^2$, $\nabla g^T(d)(\bar{d} - d) < 0$, we can find shorter distance, and this contradicts our assumption. ■

Corollary A.1 (Separate point and convex set)

Let $C \subseteq \mathbb{R}^n$ be a closed convex set and $x \in R^n$ a point not in C. Then x and C can be strictly separated by a hyperplane.

Note on Special case: convex cone *Particularly, if C is a convex cone, then we can find a horizontal plane through the origin to separate C and any point outside C, i.e., for any $x \notin C$, there exists nonzero $d \in R^n$ such that $d^T x < 0$ ($d^T y \geq 0$) for all $y \in C$.*

Definition A.7 (Supporting hyperplane)

A hyperplane containing a convex set C in one of its closed half spaces, and containing a boundary point of C.

Lemma A.5

Let C be a convex set, H a supporting hyperplane of C, and T the intersection of H and C. Every extreme point of T is an extreme point of C.

Proof Suppose there exists $x \in T$ such that x is not an extreme point of C, then it is enough to show that it is also not an extreme point of T. If so, there must exist $x_1, x_2 \in C$, $x = \alpha x_1 + (1-\alpha)x_2$. And x must belong to H (intersection), $a^T x = b = \alpha a^T x_1 + (1-\alpha) a^T x_2$. Since C is in one of H's half spaces, suppose C is in H^+, then we have $a^T x_1 \geq b, a^T x_2 \geq b$, $a^T x = b = \alpha a^T x_1 + (1-\alpha) a^T x_2 \geq \alpha b + (1-\alpha) b$. And it must be $a^T x_1 = a^T x_2 = b$. Thus, x is also not an extreme point of T. ■

Theorem A.4 (Farkas Lemma (Ali Ahmadi, 2016, Lec. 5))

Let $A \in R^{m\times n}$ and $b \in R^m$. Then exactly one of the following sets must be empty:

1. $\{x \mid Ax = b, x \geq 0\}$
2. $\{y \mid A^T y \leq 0, b^T y > 0\}$

Proof (Ali Ahmadi, 2016, Lec. 5)

(ii) to (i). Suppose there exists $Ax = b, x \geq 0$, then we have $x^T A^T y = b^T y > 0$, this contradicts our assumption.

(i) to (ii). Let $a_1, ..., a_n$ denote all columns of A, and $cone\{b_1, ..., b_n\}$ denote the cone of all non-negative combinations. Then C is convex and closed. Let $\{z_k\}$ be a sequence of points in $cone(S)$ converging to a point $\bar{z}$. Considering the following linear program:

$$\min_{\alpha,z} \|z - \bar{z}\|_\infty$$
$$\text{s.t.} \sum_{i=1}^{n} \alpha_i s_i = z$$
$$\alpha_i \geq 0$$

The objective value must be non-negative (norm), for each z_k, there exists α_k that makes the pair (z_k, α_k) feasible to the LP. As z_k get arbitrariliy close to $\bar{z}$, we conclude that the optimal value of this LP is zero. Since LP achieve their optimal values, it follows that $\bar{z} \in cone(S)$.

Suppose there exists b which cannot be represented by A, i.e., $b \notin C$. On the basis of Separating Hyperplane Theorem, the point b and the set C can be (even strictly) separated; i.e.,

$$\exists y \in \mathbb{R}^m, y \neq 0, r \in \mathbb{R} \text{ s.t. } y^T z \leq r \forall z \in C \text{ and } y^T b > r$$

Since $0 \in C$, we must have $r \geq 0$. If $r > 0$, we can replace it by $r' = 0$. For example, in the case of $y^T z > 0$, we can increase α to large enough such that $y^T(\alpha z)$ is also large enough. However, $\alpha z \in C$ contradicts Separating Hyperplane Theorem, thus,

$$y^T z \leq 0, \forall z \in C \text{ and } y^T b > 0$$

Since $a_1, ..., a_n \in C$, we see that $A^T y \leq 0$. ■

Note on *These two sets construct strong alternatives (Ali Ahmadi, 2016, Lec. 5), i.e., there is only one set is feasible. By contrast, weak alternatives means at least one set are feasible.*

This theorem is useful to prove that LP is infeasible, if (2) holds, then (1) cannot hold.

Note on Geometric interpretation *Let $a_1, ..., a_n$ denote all columns of A, and $cone\{b_1, ..., b_n\}$ denote the cone of all non-negative combinations. Then only one of two cases will hold: b is in the cone, and b is not in the cone. Thus, we can separate b and the cone with a hyperplane (Ali Ahmadi, 2016, Lec. 5).*

Theorem A.5 (Farkas Lemma (P. Williamson, 2014, Lec. 7))

Let $A \in R^{m\times n}$ and $b \in R^m$. Then exactly one of the following sets must be empty:

1. $\{x \mid Ax \leq b\}$

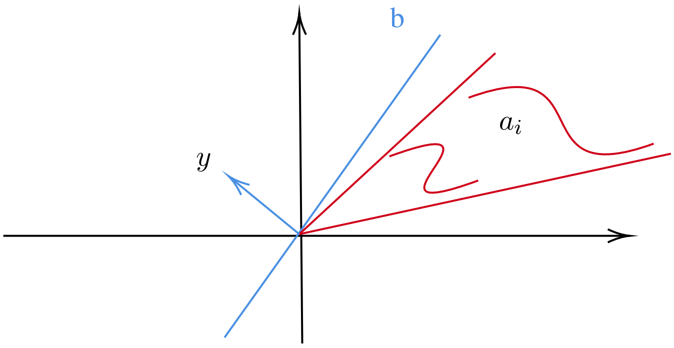

Figure A.2: Geometric interpretation of the Farkas lemma

2. $\{y \mid A^T y = 0, b^T y < 0, y \geq 0\}$

2' $\{y \mid A^T y = 0, b^T y = -1, y \geq 0\}$

Proof First we prove that (2) iff (2'). The if side is clear. If (2) is true, let $\hat{y} = -\frac{1}{y^T b} y$ and this change (2) to (2').

Secondly, we cannot have both (1) and (2). Suppose otherwise, then we have $b^T y \geq 0$ contradicts our assumption.

Now suppose (1) does not hold, so (2') does not hold either. Define a new system $A^T y = 0, y^T b = -1$ as

$$\bar{A} = \begin{bmatrix} A^T \\ b^T \end{bmatrix} \quad \bar{b} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix}$$

If (2') holds, there does not exists$z \in R^m$ such that $z \geq 0$ and $\bar{A} z = \bar{b}$. Similarly, on the basis of Separating Hyperplane Theorem, there exists s such that $\bar{A}^T s \geq 0$ and $\bar{b}^T s < 0$. Set s for $x \in R^n$ and $\lambda \in R$.

$$s = \begin{bmatrix} x \\ \lambda \end{bmatrix}$$

Then $\bar{b}^T s < 0$ implies that

$$\begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix}^T \begin{bmatrix} x \\ \lambda \end{bmatrix} < 0$$

which implies that $\lambda > 0$. Also $\bar{A}^T s \geq 0$ implies that

$$\begin{bmatrix} A^T \\ b^T \end{bmatrix}^T \begin{bmatrix} x \\ \lambda \end{bmatrix} \geq 0$$

which implies that $Ax + \lambda b \geq 0$ or that $A\left(\frac{-x}{\lambda}\right) \leq b$. Therefore $-x/\lambda$ satisfies (1), so that (1) holds. ■

Definition A.8 (Polyhedron)

Polyhedron:= $P = \{x \in \mathbb{R}^n : Ax \leq b\}, A \in \mathbb{R}^{m\times n}, m \geq n$

Definition A.9 (V-polytope, H-polytope (Toth et al., 2017, Ch. 15))

1. *V-polytope: The convex hull of a finite set* $X = \{x^1, \ldots, x^n\}$ *of points in* R^d,
$$P = \operatorname{conv}(X) := \left\{\sum_{i=1}^{n} \lambda_i x^i \mid \lambda_1, \ldots, \lambda_n \geq 0, \sum_{i=1}^{n} \lambda_i = 1\right\}$$
2. *H-polytope: The solution set of a finite system of linear inequalities with the extra condition that the set of solutions is bounded.*
$$P = P(A, b) := \left\{x \in \mathbb{R}^d \mid a_i^T x \leq b_i \text{ for } 1 \leq i \leq m\right\}$$

Note on *Polytope is a bounded polyhedron. Note that the definition in Luenberger and Ye (2015) is different from the main stream, here we adopt the definition from the main stream.*

Definition A.10 (Bounded Polyhedron)

A polyhedron P *is bounded if* $\exists M > 0$*, such that* $||x|| \leq M$ *for all* $x \in P$.

Lemma A.6 (Main Theorem of Polytope Theory)

The definitions of V-polytopes and H-polytopes are equivalent. That is, every V-polytope has a description by a finite system of inequalities, and every H-polytope can be obtained as the convex hull of a finite set of points (its vertices).

Lemma A.7 ((P. Williamson, 2014, Lec. 4))

Any polyhedron $P = \{x \in \Re^n : Ax \leq b\}$ *is convex.*

Lemma A.8 (Minkowski sum of Polytope)

Suppose that $P^i = \{x \geq 0 : A^i x = b^i\}$ *for* $i = 1, 2$ *are both bounded. Then* $P = P^1 + P^2$ *is also a polytope, where* $P^1 + P^2 = \{x^1 + x^2 : x^1 \in P^1 \text{ and } x^2 \in P^2\}$.

Proposition A.1 (Open Set and Optimality)

S *is an open set if for each* $x_0 \in S$*, there is an* $\varepsilon > 0$ *such that* $||x - x_0|| < \varepsilon$ *implies that* $x \in S$*. Show that if* S *is an open set, the problem Maximize* $\{c^T x : x \in S\}$*, where* $c \neq 0$*, does not possess an optimal point.*

Proof Suppose for the sake of contradiction that there is an optimal point x_0, we can construct another point $x_0 + \varepsilon c$, where $\varepsilon > 0$, an open feasible region means we can find a small ε to ensure $x_0 + \varepsilon c \in S$, and then show that $x_0 + \varepsilon c$ is optimal than x_0. ■

A.3 Extreme point, direction and Representation theorem

Definition A.11 (Extreme Point)

A point x in a convex set C is an extreme point of C if there are no two distinct points x_1, x_2 in C such that $x = \alpha x_1 + (1-\alpha)x_2 \in C$, for some $0 < \alpha < 1$.

Definition A.12 (Ray)

A collection of points in the form of $\{x_0 + \lambda d : \lambda \geq 0, d \neq 0\}$

Definition A.13 (Direction of the Set)

A non-zero vector d is a direction of the convex set C if for each $x_0 \in C$, the ray $\{x_0 + \lambda d : \lambda \geq 0, d \neq 0\}$ also belongs to C.

Definition A.14 (Extreme Direction)

A direction is an extreme direction of C if there are no two distinct directions d_1, d_2 such that $d = \alpha d_1 + (1-\alpha)d_2 \in C$ for some $0 < \alpha < 1$.

Theorem A.6 (Representation Theorem)

Let $X = \{x : Ax = b, x \geq 0\}$ be a non-empty set. Then the set of extreme points is non-empty and has a finite number of elements, say $x_1, ..., x_k$. The set of extreme directions is empty iff X is bounded. If X is not bounded, then the set of extreme directions is non-empty and has a finite number of elements, say $d_1, ..., d_l$. Moreever, $\overline{x} \in X$ iff it can be represented as a convex combination of $x_1, ..., x_k$ plus a non-negative linear combination of $d_1, .., d_l$, that is,

$$\bar{x} = \sum_{j=1}^{k} \lambda_j x_j + \sum_{j=1}^{l} u_j d_j, \quad \sum_{j=1}^{k} \lambda_j = 1, \lambda_j \geq 0, j = 1, \ldots, k; u_j \geq 0, j = 1, \ldots, l$$

Note on *Representation theorem shows that all solution $\overline{x}$ can be represented in this way. On the basis of this representation, we can derive the optimal solution.*

$$\begin{aligned} \min \sum_{i=1}^{n} c_i x_i = c^T x = c^T (\sum_{j=1}^{k} \lambda_j x_j + \sum_{j=1}^{l} u_j d_j) \\ \Longleftrightarrow \min_{\lambda_j, \mu_j} \sum_{j=1}^{k} \lambda_j (c^T x_j) + \sum_{j=1}^{l} u_j (c^T d_j) \\ s.t. x \in X \quad \text{feasible set} \end{aligned} \tag{A.1}$$

If feasible set is unbounded, $c^T d_j$ can be ≥ 0 or < 0. When $c^T d_j \geq 0$, it is optimal to assign $u_j = 0$. When $c^T d_j < 0$, it is optimal to assign $u_j = -\infty$ (we say the problem is unbounded).

If feasible set is bounded, then there is no such d_j, i.e., there is no extreme direction. Thus, to optimize the problem, we can find the minimal $c^T x_j$ and let $\lambda_j = 1$ and $\lambda_{i \neq j} = 0$.

Appendix Graph and Network

B.1 Notation of Graph

Definition B.1 (Graph (Network))

A network (graph) (N, A, G) *consists of*

1. $N = \{1, ..., n\}$*: the node set (*n *nodes);*
2. A*: the arcs set (*m *arcs);*
3. G*:* $n \times n$ *adjacency matrix, where* $g_{ij} = 1$ *if* i *and* j *are connected, and* $g_{ij} = 0$ *otherwise.*

Note on Network Flow *In network flow problem, the 1st node we call it source node, and the last node we call it sink node.*

Note on *Social Network Analysis In SNA, we assume that there is no self-loops, i.e.,* $g_{ii} = 0$*. For example,*

1. *Empty network:* $g_{ij} = 0$ *for all* $i \neq j$*;*
2. *Complete network:* $g_{ij} = 1$ *for all* $i \neq j$*;*
3. *Directed network: there exists* g_{ij} *s.t.* $g_{ij} \neq g_{ji}$*;*
4. *Weighted network: there exists* g_{ij} *s.t.* $0 < g_{ij} < 1$*.*

Definition B.2 (Subgraph)

Suppose $S \subseteq N$ *is the subset of nodes set, then a subgraph based on* S *is defined as*

$$G|_S = \{ij \mid g_{ij} = 1 \text{ and } i, j \in S\}.$$

Definition B.3 (Degree)

Degree(i) is the total number of arc that are incident to node i.

$$indegree(i) + outdegree(i) = degree(i)$$

Note on *Network flow We can always find a node with indegree=0 (source node) and a node with outdegree=0 (sink node).*

Lemma B.1

The sum of degrees is equal to twice the number of arcs, i.e. $\sum_{i \in N} d_i(G) = 2m$*.*

Definition B.4 (Walk)

A walk in G *is a* G' *consisting of nodes and arcs* $i_1 - a_1 - i_2 - ... - i_{r-1} - a_{r-1} - i_r$ *satisfying that* $a_k = (i_k, i_{k+1}) \in A$ *or* $a_k = (i_{k+1}, i_k) \in A$*.*

Definition B.5 (Path)

A path is a walk without repetition of nodes. A walk is more free and a path can not go back to points visited before.

Definition B.6 (Loop)

A loop is an arc whose tail node is the same as its head node.

Definition B.7 (Cycle)

A cycle is a path $i_1 - i_2 - ... - i_{r-1} - i_r$ *together with arc* (i_r, i_1) *or* (i_1, i_r).

Proposition B.1 (Degree and cycle (Kanti, 2018))

A graph with all degrees greater than 2 contains a cycle.

Definition B.8 (Geodesic)

A geodesic between i *and* j *is a shortest path between* i *and* j.

Definition B.9 (Component)

A component of (N, G) *is its maximal connected subnetwork:* (N', G') *s.t.*

1. *subnetwork:* $\emptyset \neq N' \subseteq N$ *and* $\mathrm{G}' \subseteq G$;
2. *connectness:* (N', G') *is connected;*
3. *maximal: if* $i \in N'$ *and* $ij \in G$, *then* $j \in N'$ *and* $ij \in G'$.

Note on *The set of components of a network* (N, g) *given* N *is denoted* $C(g)$, *and the component containing a specific node* i *is denoted* $C_i(g)$. *Let* $\Pi(N, g)$ *denote the partition of* N *induced by the network* (N, g). *For example, p48. In other words, a network is connected iff it consists of a single component, i.e.,* $\Pi(N, g) = \{N\}$.

Definition B.10 (Bridge)

A link ij *is a bridge in the network* g *if* $g - ij$ *has more components than* g.

Definition B.11 (Acyclic graph)

A graph is acyclic if it contains no directed cycle.

Definition B.12 (Connectivity and Strong Connectivity)

A network is connected if every two nodes are connected by some path in the network. A connected graph is strongly connected if it contains at least one directed path from one node to every other node.

Definition B.13 (Cut, s-t Cut)

A cut is a partition of the nodes in N *into two parts:* S *and* $\bar{S} = N - S$.

B.2 Special Network

Definition B.14 (Tree Network)

A tree is a connected graph that contains no cycles.

Lemma B.2

- *A tree on n nodes contains $n-1$ arcs.*
- *A tree has at least two leaf nodes (degree=1).*
- *Every two nodes in a tree are connected by a unique path.*

Definition B.15 (Forest)

A forest is a network such that each component is a tree. Thus any network that has no cycles is a forest.

Definition B.16 (Cayley Tree)

Each node besides leaves has degree d.

Definition B.17 (Rooted tree)

A rooted tree is a tree with a specifically designated node, called its root. Arcs in a rooted tree define predecessor-successor relationships.

Definition B.18 (Direct-out tree)

A direct-out tree is a tree rooted in s, where the unique path in the tree from node s to every other node is a directed path.

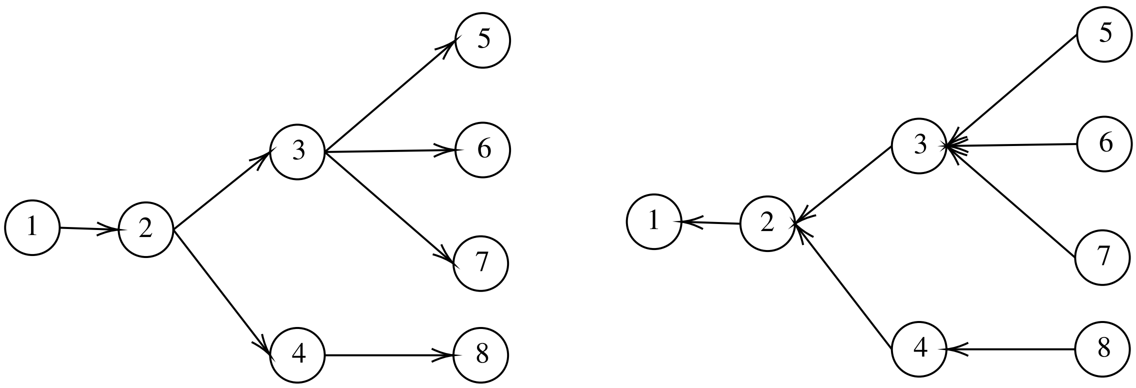

Figure B.1: Instances of directed out-tree and directed in-tree.

Definition B.19 (Spanning tree)

A tree T is a spanning tree of G if T is a spanning subgraph of G.

Theorem B.1

Let $G = (N, \epsilon)$ be a connected undirected graph and let ϵ_0 be some subset of the set ϵ of arcs. Suppose that the arcs in ϵ_0 do not form any cycles. Then, the set ϵ_0 can be augmented to a set $\epsilon_1 \supset \epsilon_0$ so that (N, ϵ_1) is a spanning tree.[a]

[a] Bertsimas (1997). Introduction to linear optimization

Definition B.20 (Fundamental cycle)

Addition of any non-tree arc to the spanning tree T creates exactly one cycle–fundamental cycle. Since a graph has $m - n + 1$ non-tree arcs, it has $m - n + 1$ fundamental cycles.

Definition B.21 (Star Network)

(N, G) is a star network if it is a tree that has a "center" node i such that every link in the network involves node i.

Note on *Star network is a special case of tree.*

Definition B.22 (Complete Network)

The complete network is a graph where all possible links are present, i.e. $g_{ij} = 1$ for all $i \neq j$.

Definition B.23 (Neighborhood (friends))

1. *The neighborhood (friends) of node i is the set of nodes that i is linked: $N_i(G) = \{j : g_{ij} = 1\}$;*
2. *All the nodes that can be reached from i by paths of length no more than k is the k-neighborhood of i, i.e.*

$$N_i^k(g) = N_i(g) \cup \left(\bigcup_{j \in N_i(g)} N_j^{k-1}(g) \right);$$

3. *Given a set of nodes S, the neighborhood of S is the union of the neighborhoods of its members, i.e.*

$$N_S(g) = \bigcup_{i \in S} N_i(g) = \{j : \exists i \in S, g_{ij} = 1\}.$$

Definition B.24 (Circle Network)

(N, G) is a circle network if it has a "single" cycle and each node in the network has exactly two neighbors.

B.3 Statistics of Network

Definition B.25 (Degree distribution)

The degree distribution of a network is a description of the relative frequencies of nodes that have different degrees.

Definition B.26 (Distance, diameter, Average path length)

1. *The distance between two nodes is the length of the shortest path or geodesic between them.*
2. *The diameter of a network is the largest distance between any two nodes in the network.*
3. *Average path length is the average taken over geodesics.*

Definition B.27 (Cliquishness)

$S \subseteq N$ *is clique if*

1. $G|_S$ *is a complete network;*
2. *for any* $i \in N \backslash S$, $G|_{S \cup \{i\}}$ *is not complete.*

Note on *A clique is a maximal completely connected subnetwork.*

Definition B.28 (Clustering measure)

If i *is a friend of* j *and* k, *CL measures how likely is it on average* j *and* k *are friend:*

$$CL(G) = \frac{\sum_i \# \{jk \in G \mid j \neq k, j, k \in N_i(G)\}}{\sum_i \# \{jk \mid j \neq k, j, k \in N_i(G)\}}$$

Note on *The most common way to measure some aspect of cliqueshness.*

Definition B.29 (Degree centrality)

Degree centrality of i *is* $\frac{1}{n-1} \sum_{j \in N} g_{ij}$.

Note on *Clustering measure is in macro level, and degree centrality is in micro level. It tells us how well a node is connected, in terms of direct connections.*

Definition B.30 (Decay centrality)

Decay centrality of i *is* $\sum_{j \neq i} \delta^{l(i,j)}$, *where* $\delta \in (0, 1)$ *is the decay parameter, and* $l(i, j)$ *is the number of links in the shortest path between* i *and* j.

Definition B.31 (Betweenness centrality)

Betweenness centrality of i *is* $\sum_{k \neq j : i \notin \{k,j\}} \frac{P_i(kj)/P(kj)}{(n-1)(n-2)/2}$, *where* $p_i(kj)$ *is the number of geodesics between* k *and* j *that involves* i, *and* $P(kj)$ *is the number of geodesics between* k *and* j.

B.4 Network Representation

Definition B.32 (Node-Arc incidence matrix)

- *Arc (i, j) has +1 in row i and -1 in row j.*
- *Storage efficiency $\frac{2m}{nm}$ and becomes inefficient when n is large.*
- *Sum of all rows is zero, it means these rows are not independent.*

Let us now focus on the ith row of A (node-arc incidence matrix), denoted by a_i'. Thus

$$\mathbf{a}_i'\mathbf{f} = \sum_{j \in O(i)} f_{ij} - \sum_{j \in I(i)} f_{ji} = b_i \quad \text{or} \quad Af = b$$

Definition B.33 (Node-Node adjacency matrix)

x

Proposition B.2 (Power of Adjacency matrix)

Let A denotes the node-node adjacency matrix of a network G. Then the ijth entry of the matrix $A^k, k = 1, \ldots, n$ is the number of walks of length k from node i to node j.[a]

[a] Finding path-lengths by the power of Adjacency matrix of an undirected graph

Proof By induction, ■

Proposition B.3 (Adjacency matrix and strongly connected[1])

Let A denotes the node-node adjacency matrix of a network G. Then G is strongly connected iff the matrix $A + A^2 + \ldots + A^n$ has no zero entry.

Definition B.34 (Forward Star Representation)

Definition B.35 (Backward Star Representation)

B.5 Hall's theorem and Bipartite Graphs

B.6 Set coverings and Independent Sets

Definition B.36 (Independent set)

An independent set of nodes $A \subseteq N$ is a set such that if $i \in A$, $j \in A$, and $i \neq j$, then $ij \notin G$.

Definition B.37 (Maximal Independent set)

An independent set of nodes A is maximal if it is not a proper subset of any other independent set of nodes.

Lemma B.3

Consider a network (N,g) and a network (N,g') such that $g \subset g'$. Any independent set A of g' is an independent set of g, but if $g' \neq g$ then there exist (maximal) independent sets of g that are not (maximal) independent sets in g'.

B.7 Colorings

B.8 Eulerian Tours and Hamiltonian Cycles

Definition B.38 (Eulerian Cycles)

x

Theorem B.2 (Eulerian Cycles theorem)

A graph (N, G) has an Eulerian cycle iff each node must have an even degree.

Proof

■

Definition B.39 (Hamiltonian cycle)

Lemma B.4 (Hamiltonian cycle theorem)

Proof

■

Bibliography

Ahuja, Ravindra, Thomas Magnanti, and James Orlin (Feb. 1993). *Network Flows: Theory, Algorithms, and Applications*. English. 1st edition. Englewood Cliffs, N.J: Pearson. ISBN: 978-0-13-617549-0.

Ali Ahmadi, Amir (2016). *ORF 523 Convex and Conic Optimization*. Princeton University. URL: https://www.princeton.edu/~aaa/Public/Teaching/ORF523/.

Bertsimas, Dimitris, John N. Tsitsiklis, and John Tsitsiklis (Feb. 1997). *Introduction to Linear Optimization*. English. unknown edition. Belmont, Mass: Athena Scientific. ISBN: 978-1-886529-19-9.

Borradaile, Glencora (n.d.). *Dijkstra's algorithm: Correctness by induction*. Oregon State University. URL: https://web.engr.oregonstate.edu/~glencora/wiki/uploads/dijkstra-proof.pdf.

Kanti, John (Dec. 2018). *Proof that if all vertices have degree at least two then G contains a cycle*. Forum post. URL: https://math.stackexchange.com/q/1335183 (visited on 06/20/2022).

Luenberger, David G. and Yinyu Ye (July 2015). *Linear and Nonlinear Programming*. English. 4th ed. 2016 edition. New York, NY: Springer. ISBN: 978-3-319-18841-6.

Malik, K., A. K. Mittal, and S. K. Gupta (Aug. 1989). "The k most vital arcs in the shortest path problem". en. In: *Operations Research Letters* 8.4, pp. 223–227. ISSN: 0167-6377. DOI: 10.1016/0167-6377(89)90065-5. URL: https://www.sciencedirect.com/science/article/pii/0167637789900655 (visited on 06/21/2022).

Orlin, James (2010). *15.082J/6.855J/ESD.78J Network Optimization*. Massachusetts Institute of Technology. URL: https://ocw.mit.edu/courses/15-082j-network-optimization-fall-2010/.

P. Williamson, David (2014). *ORIE 6300 Mathematical Programming I*. Cornell University. URL: https://people.orie.cornell.edu/dpw/orie6300/.

Toth, Csaba D., Joseph O'Rourke, and Jacob E. Goodman, eds. (Nov. 2017). *Handbook of Discrete and Computational Geometry*. English. 3rd edition. Boca Raton: Chapman and Hall/CRC. ISBN: 978-1-4987-1139-5. URL: http://www.csun.edu/~ctoth/Handbook/HDCG3.html.