Linear Programming Review Material

Jianing Yao
Department of MSIS-RUTCOR
Rutgers University, the State University of New Jersey
Piscataway, NJ 08854 USA

April 6, 2015

1 Basic Simplex

1.1 Preliminaries

Before running to the algorithm, let's first introduce some facts:

Definition 1.1 A point x is an extreme point if no other points $x^1, x^2 \in X$ exist such that

$$x = \frac{1}{2}x^1 + \frac{1}{2}x^2$$

Theorem 1.1 A convex and compact set in $\mathbb{R}^n$ equals to the convex hull of its extreme points.

This result above is essential for *linear programming*:

Theorem 1.2 Let $A \in \mathbb{R}^{m \times n}$ and let $X \subset \mathbb{R}^m$ be defined as:

$$X = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$$

A point x is an extreme point of X if and only if the columns of A that correspond to positive component of x are linearly independent.

Those extreme points are called *basic feasible solution*, their role is evident:

Theorem 1.3 If the set $X = \{x \in \mathbb{R}^n : Ax = n, x \geq 0\}$ is bounded it is the convex hull of the set of feasible solutions.

We now define the *level set:*

$$M_\beta = \{x \in \mathbb{R}^n : f(x) \leq \beta\}$$

Lemma 1.4 If $f : \mathbb{R}^n \mapsto \mathbb{R}$ is convex, then for each $\beta \in \mathbb{R}$, M_β is convex.

Lemma 1.5 Let $X \subset \mathbb{R}^n$ be a convex set and $f : \mathbb{R}^n \mapsto \mathbb{R}$ be a convex function. Then the set $\bar{X}$ of solutions of the optimization problem:

$$\min_{x \in X} \; f(x)$$

is convex.

Theorem 1.6 Let f be a concave function and let $X \subset \operatorname{dom} f$ be a convex, closed and bounded set. Then the set of solutions of the problem:

$$\min_{x \in X} \; f(x) \tag{1}$$

contains at least one extreme point of X. If, in addition, the function $f(\cdot)$ is affine, the set of solutions of (1) is the convex hull of the set of extreme points of X that are solutions of (1).

Proof. Suppose $\hat{x}$ is the optimal solution of 1, since X is convex and compact, we can find extreme points $x^1, x^2, ..., x^m$ of X such that

$$\hat{x} = \sum_{i=1}^{m} \alpha_i x^i$$

where $\sum_{i=1}^{m} \alpha_i = 1$. By concavity of the $f(\cdot)$,

$$f(\hat{x}) \geq \sum_{i=1}^{m} \alpha_i f(x^i)$$

We can rewrite it as:

$$0 \geq \sum_{i=1}^{m} \alpha_i [f(x^i) - f(\hat{x})]$$

Since $f(\hat{x}) \leq f(x^i)$, $i = 1, ..., m$, in fact,

$$f(\hat{x}) = f(x^i)$$

Thus, we can draw the conclusion that the set of solution to (1) is included in the convex hull of the extreme points that are solutions of 1.

Now if $f(\cdot)$ is affine, it is also convex, by above theorem, the solution set is convex. The assertion follows. $\square$

Application of it in linear programming gives us the following result:

Corollary 1.7 If the feasible set of the linear programming problem:

$$\begin{aligned} \min . \; & c^\top x, \\ \text{subject to: } & Ax = b, \\ & x \geq 0. \end{aligned} \tag{2}$$

is bounded, then the set of optimal solutions is the convex hull of the set of optimal basic feasible solutions.

The above fact is used by simplex method for solving linear programming problems. It moves from one basic feasible solution to a better one, as long as progress is possible. The best basic feasible solution is guaranteed to be optimal. It can be found after finitely many steps if a solution exists. If the set if unbounded, we may discover a ray from the recession cone, along which objective can be decreased without limits. In this case, no optimal solution exists.

1.2 Simplex Method – Matrix Form

Although linear programming can appear in different form, it can be easily converted so called *standard form* as in (8), for convenience, we restate it here:

$$\begin{aligned} \min . \ & c^\top x, \\ \text{subject to: } & Ax = b, \\ & x \geq 0. \end{aligned} \tag{3}$$

Here, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$. Let's make an assumption that we have a initial basic feasible solution, later on we will justify how to find such solutions. To illustrate the procedure of simplex, we first partition the matrix A and index set into two parts, I_B, I_N and

$$\begin{bmatrix} B & | & N \end{bmatrix}$$

where B corresponds to the basic feasible solution (in I_B) that we assume to have already and N (in I_N) corresponds to non-basic feasible solution. Accordingly, we partition x and c as well:

$$x = [x_B \quad x_N]^\top, \ c = [c_B \quad c_N]^\top$$

We also define the value function $z := c^\top x$. Under this convention, we have

$$Bx_B + Nx_N = b \tag{4}$$

In the first iteration, we get $x_B = B^{-1}b$ (since non-basic variable are set to be 0) and the value function is:

$$z = c_B^\top x_B + c_N^\top x_N \tag{5}$$

From (4),

$$x_B = B^{-1}b - B^{-1}Nx_N$$

Plug into (5),

$$\begin{aligned} z &= c_B^\top x_B + c_N^\top x_N \\ &= c_B \top (B^{-1}b - B^{-1}Nx_N) + c_N^\top X_N \\ &= c_B^\top B^{-1}b + (c_N^\top - c_B^\top B^{-1}N)x_N \end{aligned}$$

We set $T = B^{-1}N$ and

$$s_N^\top = c_N^\top - c_B^\top T$$

Our objective is to reduce the value function z. If some coordinates of $s_N^\top$ is negative, i.e., $(s_N^\top)_j < 0$, for some $j \in I_N$, we can increase corresponding $(x_N)_j$ so that z can be reduced. Take $\alpha > 0$, $(x_N)_j$ is increased from 0 to α. This means that we can bring one of the non-basic variable in to the basis. But the next question is to remove which one out of the basis. Let's go back to the equation (4), notice $x_N = \alpha e^j$,

$$x_B = B^{-1}b - \alpha T e_j$$

For each $i \in I_B$,

$$\begin{aligned}(x_B)_i^{\text{new}} &= (B^{-1}b)_i - \alpha e_i^\top T e_j \\ &= (x_B)_i^{\text{old}} - \alpha T_{ij}\end{aligned}$$

The restriction on α is that we can not make $(x_B)_i^{\text{new}}$ negative, thus

$$\alpha \leq \min_{i \in I_B} \frac{(x_B)_i^{\text{old}}}{T_{ij}} \tag{6}$$

Of course, when $T_{ij} < 0$, we don't need to worry about it. Thus, the index $i \in I_B$ at which the minimum attained is the basic variable which leaves the basis. After one iteration, $I_B^{\text{new}} = I_B \setminus \{i\} \cup \{j\}$, $I_N^{\text{new}} = I_N \setminus \{j\} \cup \{i\}$, and we redo the process again and again until $(s_N^\top)_k \geq 0$ for all $k \in I_N$.

Let's now find an initial basic feasible solution. Consider the following auxiliary problem:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n-m} x'_i, \\ \text{subjec to}: \quad & Ax \pm x' = b, \\ & x, x' \geq 0. \end{aligned} \tag{7}$$

Note, if $b \geq 0$, we use $+$, otherwise, we use $-$. For this problem, we get a free basic feasible solution: $x = 0$, for $i = 1, ..., n$, $x'_i = |b|$ for $i = 1, ..., m - n$. We run the simplex on (7), if $x' = 0$, $x = \bar{x}$, then $\bar{x}$ is a basic feasible feasible solution for the original problem.

What if, in the optimal solution, some of x''s are positive? This implies the original problem is infeasible. Because if $\bar{x}$ is feasible for the original problem, then setting $x' = 0$ would be feasible to the auxiliary problem with objective function value 0, which is better than the current minimum.

Remark 1.8 The auxiliary problem can not unbounded because we know that $x' = 0$ and the value of 0 is a lower bound for the objective. Also, since $x = 0$, $x' = |b|$ is feasible, $\sum_{i=1}^{n-m}$ is an upper bound on the value of objective function.

1.3 Infeasibility, unboundedness, degeneracy and revised simplex method

1. Infeasibility: As we already discussed in the last subsection, the auxiliary problem will detect the infeasibility of the original problem.

2. Unboundedness: The unboundedness means that we can decrease the value function indefinitely many, e.g., $\alpha = +\infty$. This happens if, $(s_N^\top)_i < 0$ and there is no restriction on (6), i.e., $T_{ij} \leq 0$ for all $i \in I_B$.

3. Degeneracy: In general, if a basic feasible solution x_B contains a 0 coordinate, i.e., $(x_B)_i = 0$ for some i, we say it is degenerate. It may cause some problems. In particular, we make zero progress by removing them. Thus, we may encounter many pivots where we are moving from one degenerate basic feasible solution to another without any improvements. Worse yet, we could get stuck in a cycle.

We shall point out, there is a simple rule to avoid cycling from happening, this is so called – *Bland's pivoting rule*:

1. if $(s_N^\top)_i < 0$ for some $i \in I_N$, we choose the one with smallest i;
2. If in the column i, there are several indices j such that $\frac{(x_B)_j^{\text{old}}}{T_{ij}}$ equal to the minimum, we choose the smallest j.

The revised simplex method delays computation of columns of $T = B^{-1}N$ until it actually needs it. Let's recall the step of calculating reduced cost vector:

$$s_N^\top = c_N^\top - c_B B^{-1} N := c_N^\top - y^\top N$$

where $y = c_B B^{-1}$. We do not compute $T = B^{-1}N$ and then $c_B^\top T$, instead, we first compute $y^\top = c_B^\top B^{-1}$, it takes $o(m^2)$ operation if we have LU factorization of B. Then, compute $y^\top N$, where N is huge but only requiring $o(m \times n)$ operations. Also, when we find $(s_N)_j < 0$, we only need to compute:

$$t_j = B^{-1} N_j$$

which again will cost $o(m^2)$, since we already have factorization of B in hand.

In summary, the computational complexity of the revised *simplex method* is the following:

1. The initial LU factorization at the first pivot: $o(m^3)$;
2. The work involved in each subsequent pivot: $o(m^2) + o(mn)$;
3. The total number of pivots. Let's call this number M.

So the total amount of operation is

$$o(m^3) + o(M(m^2 + mn))$$

The mystery number is M. In the worst case this number can be very large, see *Klee-Minty example.*

2 Duality, Complementarity and Farkas Lemma

2.1 Duality Relation

For standard problem:

$$\begin{aligned} \min.\ & c^\top x, \\ \text{subject to: }\ & Ax = b, \\ & x \geq 0. \end{aligned} \tag{8}$$

Let's consider the *Lagrangian*, with multipliers $y \in \mathbb{R}^m$,

$$L(x, y) = c^\top x + y^\top (Ax - b) \tag{9}$$

Then, the dual function is

$$L_D(y) = \min_{x \geq 0} L(x, y) = \min_{x \geq 0} \left[(c^\top + y^\top A)x - b^\top y \right]$$

To have this value finite, we shall require $c^\top + y^\top A \geq 0$. Thus, the dual problem is:

$$\begin{aligned} \max.\ & -b^\top y, \\ \text{subject to: }\ & c^\top + y^\top A \geq 0. \end{aligned} \tag{10}$$

Since $y \in \mathbb{R}^m$, we can rewrite as:

$$\begin{aligned} \max.\ & b^\top y, \\ \text{subject to: }\ & y^\top A + s = c, \\ & s \geq 0 \end{aligned} \tag{11}$$

where s is the dual slack variable.

Lemma 2.1 *(Weak Duality)* Suppose x is feasible for the primal problem and let (y, s) be feasible for the dual problem, then $c^\top x \geq b^\top y$.

Remark 2.2 We actually can calculate the *duality gap*:

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

Or, equivalently, $x_1 s_1 + \cdots + x_n s_n \geq 0$.

This simple theorem give us several implications:

1. every value $b^\top y$ for a dual feasible y is a lower bound on the primal problem. Likewise, every value $c^\top x$ for primal feasible x is an upper bound for the dual problem. The quantity $x^\top s \geq 0$ is called the *duality gap* as we mentioned;

2. if the primal problem is unbounded, then the dual problem is infeasible. If the dual problem is unbounded, the minimization problem is infeasible;

3. suppose the duality gap is zero at $(\bar{x}, (\bar{y}, \bar{s}))$, then $\bar{x}$ is optimal for the primal and $(\bar{y}, \bar{s})$ is optimal for the dual.

The question is when both the primal and dual problems are feasible, is it true that at the optimum the duality gap is always zero? The answer is yes for linear model, and the proof comes from the simplex.

Remember that the simplex stops when

$$s_N^\top = c_N^\top - c_B^\top B^{-1} N = c_N^\top - y^\top b \geq 0$$

In the final tableau, we define $x = [x_B \ 0]$, $s = [0 \ s_N]$, $y = B^{-\top} c_B$. Now, x is primal feasible, because

$$Ax = [B \ N][x_B \ 0]^\top = Bx_B = b$$

and (y, s) is dual feasible: $s \geq 0$ and

$$A^\top y + s = [B^\top \ N^\top] y + [0^\top \ s_N^\top] = c$$

The duality gap is 0,

$$c^\top x - b^\top y = c_B^\top B^{-1} b - y^\top b = c_B^\top B^{-1} b - (c_B^\top B^{-1}) b = 0$$

Notice, during simplex algorithm, the duality gap is always 0. We actually shows the fundamental result of linear programming:

Theorem 2.3 *(Strong Duality)* For the primal-dual pair of linear programs,

1. if the primal problem is unbounded, the dual problem is infeasible; if the dual is unbounded, the primal is infeasible;

2. if both the primal and dual are feasible, then both have optimal solutions, $(x^*, (y^*, s^*))$. Furthermore, $(x^*)^\top s^*$.

In the process of proving the duality theorem, we showed that the duality gap is 0 at the end (actually, during the whole process of simplex). But we know that $x^\top s = x_1 s_1 + \cdots + x_n s_n$ is sum of a bunch of non-negative numbers. If this sum adds up to 0, it mean that each $x_i s_i = 0$. This results in the following theorem:

Theorem 2.4 *(the complementary slackness theorem)* If x^* is primal optimal and s^* dual optimal, then $x_i^* s_i^* = 0$, $i = 1, ..., n$.

Since $s_i = c_i - a_i^\top y$ where a_i is the i-th column of A, we can state the complementary slackness theorem as follows: each primal variable x_i corresponds to a dual constraint $a_i^\top y \leq c_i$. Furthermore, if (x^*, y^*) are optimal, then

1. if $x_i^* > 0$ at the optimum, then $a_i^\top y^* = c_i$;

2. if $a_i^\top y^* < c_i$ at the optimum, then $x_i^* = 0$.

because $x_i^* s_i^* = x_i^*(c_i - a_i^\top y^*) = 0$. Another way of looking at complementary slackness theorem is to observe that the following non-linear system of equations:

$$\begin{aligned} Ax &= b, \\ A^\top y + s &= c, \\ x_i s_i &= 0, \ i = 1, ..., n. \end{aligned} \tag{12}$$

This is so called *Karush-Kuhn-Tucker(KKT) condition* necessary optimality condition.

2.2 Farkas Lemma

A number of results related to duality theory are in the form of *theorems of alternatives: either statement A is true or statement B is true, but not both!*. The most famous of these is the *Farkas lemma*

Lemma 2.5 *(Farkas)* Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$, then, either (1) there exists a non-negative vector $x \geq 0$, such that $Ax = b$; or, (2) there exists a vector $y \in \mathbb{R}^n$: $y^\top A \leq 0$ and $y^\top b > 0$.

Proof. (proof without simplex) If (1) has a solution, say $\hat{x} \geq 0$ such that $A\hat{x} = b$, then

$$\langle y, b\rangle = \langle Ax, y\rangle = x^\top A^\top y \leq 0$$

which contradicts $y^\top b > 0$.

For the other way around, let's define $C = \{Ax \mid x \geq 0\}$. If (1) does not have a solution, then $b \notin C$, since C is closed and convex, there exists y such that

$$y^\top b - \epsilon \geq y^\top Ax = \sup_{x\geq 0} y^\top Ax$$

In particular, if we choose $x = 0$, we get $y^\top b > 0$. If $y^\top A \leq 0$, we can choose $\bar{x} = (\alpha > 0, 0, ..., 0)^\top$ so that

$$\sup_{x\geq 0} y^\top Ax \geq \alpha(y^\top A)_1$$

Let $\alpha \to \infty$, $\sup_{x\geq 0} y^\top Ax \to \infty$, contradicts the fact that it is bounded above by $y^\top b - \epsilon$. $\square$

Proof. (Proof by simplex method and duality) Consider

$$\min 0^\top x \text{ such that } Ax = b, x \geq 0 \tag{13}$$

$$\max b^\top y \text{ such that } A^\top y \leq 0 \tag{14}$$

If "either" part is true, then the minimization problem is feasible and has optimal value 0. By weak duality, for any feasible dual, $A^\top y \leq 0$, $b^\top y \leq o^\top x = 0$, so or part is false.

If "or part" is true, then it is also true that

$$A^\top(\mu y) \leq 0, \ b^\top(\mu y) > 0 \tag{15}$$

This means that the dual problem is unbounded, therefore, the primal is infeasible. The "either part" is false. $\square$

Remark 2.6 We have shown that weak duality + simplex $\Rightarrow$ strong duality $\Rightarrow$ Farkas lemma. Actually, we can also show that weak duality + Farkas Lemma $\Rightarrow$ strong duality $\Rightarrow$ correctness of simplex.

Let's sketch the proof. Suppose the optimal value of the primal problem is $z = c^\top x$, then $[A; \ -c^\top]x = [b; -z-\epsilon]$, for $\epsilon > 0$ has no solution. By Farkas lemma, there exists $\bar{y} = [y \ \alpha]$ such that $a_i^\top y \le \alpha c_i$ for all i, and $b^\top y > \alpha(z+\epsilon)$. If we scale $\bar{y}$ to make $\alpha = 1$, then

$$a_i^\top y \le c_i \text{ and } b^\top y > z + \epsilon \tag{16}$$

Since, by weak duality, $b^\top y \le z$, we have $b^\top y = z$ as $\epsilon \to 0$.

Let's look at an interesting application of Farkas lemma in Finance. Consider a situation where you bet on one of m choice. After you make your wage, one of n events happens. There is a pay-off matrix $R \in \mathbb{R}^{m\times n}$. If you wage 1 dollar on choice i and event j happens, your reward will be R_{ij}. The question is: can you make a combination of wagers so that you will always have a total positive gain? If we call the vector $g^\top = x^\top R$ *gain vector*, we want to know whether there exists a choice such that $g^\top > 0$. Such situation is called *arbitrage.*

Theorem 2.7 Given wagers $1, 2, ..., m$, and outcome $1, 2, ..., n$, and the pay-off matrix R_{ij}, $i = 1, ..., m$, $j = 1, ..., n$, then, either, (1) there is a combination of wagers x such that $x^\top R > 0$; or, (2) there exists a probability vector p, $p \ge 0$, $\mathbf{1}p = 1$ such that $Rp = 0$.

Proof. The arbitrage theorem is a rewording of the Farkas lemma where 'either' and 'or' are switched. In Farkas lemma, set $A = [R \ \mathbf{1}^\top]^\top$ and $b = [0 \ 1]^\top$, then the 'or' says: $\exists p$, $Ap = b$, $p \ge 0$. If this is false, by Farkas lemma, $\exists y$ such that $y^\top A = R^\top x + x_0\mathbf{1} \ge 0$ and $y^\top b < 0$. From the latter one, we know that $x_0 < 0$. However, this means that $R^\top x \ge -x_0\mathbf{1} > 0$. Thus, arbitrage opportunity exists. □

2.3 Dual Simplex and Primal-dual Methods

Let's ask the following question, how does the simplex method looks like from the point of view of the dual problem? When running the simplex, from primal problem, we always have $x_B = B^{-1}b$, where $x \ge 0$, obviously, x is feasible but not optimal. For dual problem,

$$y = c_B^\top B^{-1}, \ s = c - A^\top y = \begin{bmatrix} c_B^\top - c_B^\top B^{-1}B & s_N \end{bmatrix} = \begin{bmatrix} 0 & s_N \end{bmatrix}$$

Here, s maybe less than 0 for some coordinates so that it is infeasible, however, the duality gap is always 0, i.e.,

$$s^\top x = c^\top x - b^\top y = 0$$

In summary, from the point of view of the primal, at each iteration, we are moving from one solution to another, maintaining feasibility $x \ge 0$ while trying to achieve optimality: making $s_N \ge 0$; from the point of dual, we are moving from one infeasible point (y, s) to another, maintaining optimality $x^\top s = 0$ while trying to achieve feasibility $x \ge 0$. So what happens if we apply the simplex method to the dual problem, and then view it from the point of view of the primal problem? That is: (1) maintain feasibility in the dual (y, s), i.e., $s \ge 0$; and (2) maintain optimality in the dual, $x^\top s = 0$ and try to move towards feasibility $x \ge 0$. Doing so we get the *dual simplex method:*

1. at each iteration, we have optimality satisfied: $s_N \geq 0$ but feasibility may not be satisfied: $x_B \not\geq 0$. Thus, instead of basic feasible solution, we simple have basic solution;

2. To improve the current solution we first choose an index i such that $(x_B)_i < 0$ to leave the basis;

3. Then we must choose an index j to enter the basis. But this must be done in a way such that $s_N = c_N - N^\top y \geq 0$. Since $y = B^{-\top} c_B$ changes, this implies that the new y, say y', is

$$y' = y - \alpha B^{-\top} e_i$$

However, this should be done in a way to maintain $s_N \geq 0$,

$$\begin{aligned} s_N^{(new)} = c_N - Ny' &= c_N - N(y - \alpha B^{-\top} e_i) \\ &= c_N - Ny + \alpha(NB^{-\top})e_i \\ &= c_N - Ny + \alpha T_i \end{aligned}$$

This implies

$$\begin{aligned} &(s_N)_j + \alpha T_{ij} \geq 0 \ \forall j \\ \Leftrightarrow &(s_N)_j \geq -\alpha T_{ij} \ \forall j \end{aligned}$$

If $T_{ij} \geq 0$, this is satisfied for all $\alpha > 0$; if $T_{ij} < 0$, then $\alpha < \frac{(s_N)_j}{-T_{ij}}$. Thus, the rule is: the column to enter the basis is the j, where $\alpha = \min_{j; T_{ij} < 0} \frac{(s_N)_j}{-T_{ij}}$;

4. If all $T_{ij} \geq 0$, $j = 1, ..., m$, then the dual is unbounded and the primal is infeasible.

Remark 2.8 The dual simplex method, in general, does not have an advantage or disadvantage over the simplex method. In special cases, where a basic dual feasible solution is available it may be better to use the dual simplex method. For example, if $c \geq 0$, then $y = 0$ is feasible since $A^\top y \leq c$ is satisfied. Another important context when the dual simplex method is useful is when a problem is solved to optimality, and then a new constrained is added making the current optimal solution infeasible. For this case, however, the current dual solution is feasible. To find the optimal solution again we may start from the current feasible dual and run the dual simplex method. In many cases, a few iterations of the dual simplex method yields the new optimal solution.

2.4 The Primal-Dual Algorithm

In addition to the simplex and the dual simplex methods, there are a number of different algorithm with the strategy to move both primal and dual solutions (possibly violating constraints) towards both feasibility and optimality. the optimality part is often achieved by pushing primal and dual solution in a way to satisfy complementary slackness conditions. These types of algorithm are called *primal-dual algorithms*.

Again, these algorithms do not have any particular advantage over the ordinary simplex, or the dual simplex method. However, in some special linear programming arising from network problems, they will be advantageous.

Consider, as usual, the standard primal and dual pair of linear programs:

$$\begin{aligned} \min\ & c^\top x \\ \text{subjec to: } & Ax = b, \\ & x \geq 0 \end{aligned}$$

and

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

Suppose y is dual feasible point, not necessarily basic or any other conditions. The set of inequalities in $A^\top y \leq c$ can be partitioned into two groups. Let $A \in \mathbb{R}^{m\times n}$

1. The set $Q \subseteq \{1, 2, ..., n\}$, where if $i \in Q$, then $a_i^\top y = c_i$;
2. the complement $Q^{\mathcal{C}}$ where if $i \in Q^{\mathcal{C}}$ then $a_i^\top y < c_i$.

From our discussion in duality, we know that each inequality constraint $a_i^\top y \leq c_i$ corresponds to a primal variable x_i. Now, if $i \in Q^{\mathcal{C}}$, and if y were dual optimum, then by complementary slackness, $a_i^\top y < c_i$ implies $x_i = 0$. On the other hand, for $i \in Q$, $a_i^\top y = c_i$ so x_i can be a non-negative number. Suppose, given the dual feasible solution y, we set all primal variables $x_i = 0$ for $i \in Q^{\mathcal{C}}$, and then try to find a basic feasible solution for the rest of x_i using a *phase I* approach. In other words, we should solve the following LP:

$$\begin{aligned} \min\ & x'_1 + x'_2 + \cdots + x'_m \\ \text{subject to: } & A_Q x_Q + x' = b \\ & x_Q, x' \geq 0 \end{aligned}$$

where $x_Q = [x_{i_1}, .., x_{i_1}]$ and $A_Q = [a_{i_1}, ..., a_{i_q}]$, if $Q = \{i_1, ..., i_q\}$ (A_Q is composed of A corresponding to Q). The linear program above is called the *restricted primal (RP)* its dual is called the *restricted dual (RD)*:

$$\begin{aligned} \max.\ & b^\top y' \\ \text{subject to: } & A_Q^\top y' \leq 0 \\ & y' \leq 1 \end{aligned}$$

We proceed by solving the above restricted primal. One of two things can happen:

1. *Case I:* the optimal solution to RP has the optimal objective value equal to 0, that is $x' = 0$, then we can safely claim that: (1) the current feasible y is dual optimal; (2) the vector $x = [x_Q\ x_{Q^c}]^\top$ where $X_{Q^c} = 0$ and x_Q the solution of RP is primal optimal. This follows from complementary slackness theorem where $x_i(c_i - a_i^\top y) = 0$ for all $i = 1, ..., n$ and feasibility of both x and y;

2. *Case II:* The optimal solution to RP is strictly positive and thus, the solution $x = [x_Q \; x_{Q^c}]^\top$ is not feasible for the primal. This also implies that our current dual solution y is not optimal and we have to change it. Suppose y' is the optimal solution to restricted dual (RD). Then we know: (*)$A_Q^\top y' \leq 0, y' \leq 1$. Also, our current feasible dual satisfies: (**) $A_Q^\top y = c_Q, A_{Q^c}^\top y < c_{Q^c}$. Now, for $\alpha \geq 0$, consider the vector $y + \alpha y'$. Then from (*) and (**), $A_Q^\top (y + \alpha y') \leq c_Q$, we would like to have

$$A_{Q^c}^\top (y + \alpha y') \leq c_{Q^c} \text{ (***)}$$

since $\alpha \geq 0$, we may choose the largest α such that (***) is satisfied. Note that for any $\alpha > 0$, this new y improves the dual objective: since y' is the optimal solution to DR $b^\top y' = \sum x_i' > 0$. Thus, $b^\top (y + \alpha y') \geq b^\top y$ for all $\alpha \geq 0$. All that's left is to determine α. We must choose $\alpha \geq 0$ so that (***) is satisfied:

$$A_{Q^c}^\top (y + \alpha y') \leq c_{Q^c}$$

implies for each $i \in Q^{\mathcal{C}}$, $c_i - a_i^\top y' \geq \alpha a_i^\top y'$. If $a_i^\top y' \leq 0$, the this is true for any α. In particular, if for every $i \in Q^{\mathcal{C}}$, $a_i^\top y' \leq 0$, then we can choose α arbitrarily large; the dual is unbounded and the primal is infeasible. If, on the other hand, $a_i^\top y' > 0$, then

$$\alpha \leq \frac{c_i - a_i^\top y}{a_i^\top y'}$$

Thus, we may choose

$$\alpha = \min_{i; a_i^\top y' > 0} \frac{c_i - a_i^\top y}{a_i^\top y'}$$

Let's summarize the *primal-dual algorithm:*

- Set $Q = \{i | 1 \leq i \leq n, a_i^\top y = c_i\}$, $Q^{\mathcal{C}} = \{1, ..., n\} \setminus Q$;
- Solve the restricted RP;
- If $x' = 0$ in the optimal RP: return $x = [x_Q \; 0]^\top$;
- If $(a_i^\top y) \leq 0$ for all $i \in Q^{\mathcal{C}}$: return dual unbounded, primal infeasible;
- set

$$\alpha \leq \frac{c_i - a_i^\top y}{a_i^\top y'}$$

where y' is the dual solution of RD;
- set $y = y + \alpha y'$;
- go to the first step.

3 Application in Game Theory and Regression

3.1 Matrix Game

Let's consider the simplest game Scissors-Rock-Paper, if we enumerate the actions, then we will have the following pay-off matrix:

$$A = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix}$$

The entry a_{ij} stands for the money that row player pays to the column player. Let y_i denote the probability that row player selects action i. The vector y composed of these probabilities is called a *stochastic vector*. Similarly, x corresponds to column player. Then

$$(1\ 0\ 0) \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = 1$$

This means when row player choose paper, column player choose scissors, row player will lose one dollar to column player.

Now, let's fix row player:

$$(1\ 0\ 0) A \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = A_{11}x_1 + A_{12}x_2 + A_{13}x_3$$

This can be interpreted as the expected gain of column player given row player chooses rock. Generally speaking:

$$e_j^\top Ax = \mathbb{E}[\text{ column player's gain}|\text{ row player choose } j\ ]$$

Suppose that the column player adopts strategy x, then the row player's best defend is to use the strategy y^* that achieves the following minimum:

$$\begin{aligned} \min\ & y^\top Ax \\ \text{subject to: } & e^\top y = 1 \\ & y \geq 0 \end{aligned}$$

Since for any given x the row player will adopt strategy that achieves the minimum above. It follows that the column player should employ a strategy x^* that achieves the following maximum:

$$\max_x \min_y\ y^\top Ax$$

which can be solved by *linear programming*. Indeed, we have already seen that the inner optimization can be taken over just the deterministic strategies:

$$\min_y\ y^\top Ax = \min_i\ e_i^\top Ax$$

Hence, we can rewrite the optimization as:

$$\begin{aligned}
&\max_x \ (\min_i \ e_i^\top Ax) \\
\text{subhject to: } &\sum_{j=1}^{n} x_j = 1, \\
&x_j \geq 0, \ j = 1, ..., n
\end{aligned}$$

Or, equivalently,

$$\begin{aligned}
\max. \ & v \\
\text{subject to: } & v \leq e_i^\top Ax, \ i = 1, 2, ..., m, \\
& \sum_{j=1}^{n} x_j = 1, \\
& x_j \geq 0, \ j = 1, 2, ..., n
\end{aligned} \tag{17}$$

In the vector notation:

$$\begin{aligned}
\max. \ & v \\
\text{subject to: } & ve - Ax \leq 0, \\
& e^\top x = 1, \\
& x \geq 0
\end{aligned}$$

In matrix form:

$$\begin{aligned}
\max \ & (0 \ 1)(x \ v)^\top \\
\text{subject to: } & (-A \ e)(x \ b)^\top \leq 0, \\
& (e^\top \ 0)(x \ v)^\top = 1, \\
& x \geq 0, \ v \text{ is free.}
\end{aligned}$$

Now, let's turn around. The row player seeks strategy y^* that attains optimality in the following *min-max* problem:

$$\min_y \ (\max_x \ y^\top Ax)$$

which can be formulate as:

$$\begin{aligned}
\min \ & u \\
\text{subject to: } & ue - A^\top y \geq 0, \\
& e^\top y = 1, \\
& y \geq 0
\end{aligned} \tag{18}$$

Theorem 3.1 There exists stochastic vector x^* and y^* for which:

$$\max_x \ (y^*)^\top Ax = \min_y \ y^\top Ax^*$$

Proof. The proof follows trivially from the observation (17) is the dual of (18). Therefore, $v^* = u^*$. Furthermore,

$$v^* = \min_i \; e_i^\top A x^* = \min_y \; y^\top A x^*$$

and similarly,

$$y^* = \max_j \; e_j^\top A^\top y^* = \max_x \; x^\top A^\top y^* = \max_x \; (y^*)^\top A x$$

$\square$

3.2 L_1-norm Regression

In regression analysis, the goal is to find the vector x that best explains the observation b, i.e.,

$$b = Ax + \epsilon$$

where ϵ has mean 0 and variance σ^2. Usually, we use so called *least square method*, which is to solve the following optimization problem:

$$\min_x \; f(x) = \min_x \; ||b - Ax||_2^2$$

Then, one can get stationary point by differentiation, since the $f(\cdot)$ is convex, hence we get the global minimum, i.e.,

$$x = (A^\top A)^{-1} A^\top b$$

Just as the median gives a more robust estimate of the average value of a collection of numbers than mean, L^1-regression is less sensitive to outliers than least squares regression is. It is defined by minimizing the L^1-norm of the deviation vector. That is to find $\hat{x}$ as follows:

$$\hat{x} = \arg\min_x \; ||b - Ax||_1$$

Unlike for least square regression, there is no explicit formula for the solution of L^1-regression problem. However, the problem can be reformulate as a linear programming problem. Indeed, it is easy to see that the L^1-regression problem:

$$\min . \sum_i |b_i - \sum_j a_{ij} x_j|$$

can be rewritten as

$$\begin{aligned} &\min . \sum_i t_i \\ &\text{subjec to: } t_i - |b_i - \sum_j a_{ij} x_j| = 0, \; i = 1, 2, ..., m. \end{aligned}$$

which is equivalent to the following linear programming problem:

$$\begin{aligned} \min \ & \sum_i t_i \\ \text{subject to: } & -t_i \le b_i - \sum_j a_{ij}x_j \le t_i, \ i = 1, 2, ..., m. \end{aligned}$$

Hence, to solve the L^1-regression problem, it suffices to solve this linear programming problem.

4 Network Flow Optimization

4.1 Basics, terminology and facts

A *directed graph or network flow* $G = (V, E)$ is composed of two sets, a finite set V of vertices or nodes and a finite set E of edges or arcs, where arcs are made up of *ordered pairs of nodes*: $E \subseteq V \times V$. In addition, there might one or more numbers attached to either arcs or node. These numbers could represent, for example, cost, distance or capacity. Formally, these numbers are represented by function: $c : E \mapsto \mathbb{R}_+$, $d : E \mapsto \mathbb{R}_+$, $s : V \mapsto \mathbb{R}$, e.t.c..

Speaking of the representation of graphs and networks, there are many ways, two most common approaches are *adjacency matrix* (usually, for undirected graph) and *incidence matrix* (directed graph), respectively. We will mainly focus on incidence matrix since we are more interested in directed network flow. In this representation, the rows are indexed by arcs in E and the columns are indexed by nodes in V. Specifically, for the row (i, j), the column i is -1 and the column j is $+1$; all other entries in row (i, j) is 0. To proceed, let's give some definitions:

Definition 4.1 Here is a list of useful definitions:

1. A sequence of arcs $i_1i_2 \to i_2i_3 \to \cdots \to i_{n-1}i_n$ going from one vertex to another is called a *walk*. It can also be represented by the sequence of nodes it visits, e.g., $i_1 \to i_2 \to \cdots \to i_n$;

2. If all the node in a walk are distinct, then the walk is called a *path*, e.g., from i_1 to i_n;

3. If all the node in a walk are distinct except the first and the last one, that is, $i_1 = i_n$, then this walk is called a *cycle*;

4. An **undirected network** $G = (V, E)$, if there is a *path* between any pair of nodes, then the network flows is called *connected*, otherwise, it is disconnected;

5. A **directed network** $G = (V, E)$ is called *weakly connected* if we disregard the direction of the arcs, it will be connected, while it is *strongly connected* if there is a *directed path* between any pair of nodes;

6. In an **undirected network**, the number of arcs connected to a vertex is called the *degree* of the vertex, while in a **directed graph**, the number of arcs (i, j) connected

to vertex i is called the *outdegree* of i and the number of arcs (j, i) connected to i is called *indegree* of i, finally sum of indegree and outdegree is simply the degree of a node;

7. A node with degree 1 is called a *leaf*;

8. An **undirected network** which is both connected and has no cycles is called a *tree*. If the network is not connected and has no cycles, it is called a *forest*. In this case, a forest's connected components are all trees.

Lemma 4.1 A tree with n nodes and at least one arc has at least two leafs. Furthermore, the number of arcs is exactly $n - 1$.

Observe that for an incidence matrix of a directed network, each row corresponds to an arc (i, j) which means there is a -1 in column i and $+1$ in column j and 0 in every other column. Thus, by adding all columns of the incidence matrix we get the vector of all 0's. In other words, if A is the incidence matrix of a directed network $G = (V, E)$, then $A\mathbf{1} = 0$.

Suppose a directed network is a tree if direction of arcs are disregarded, the incidence matrix of such a tree will also have the property that sum of its columns is 0. In addition, such a matrix has $b = |V|$ columns and $n - 1$ rows. Suppose we remove a column corresponding to a leaf, then we will have an $(n - 1) \times (n - 1)$ square matrix made up of 0's, 1's and -1's. Let's call this matrix the *truncated incidence matrix of* G. The following fact gives a fundamental property of trees' incidence matrix":

Theorem 4.2 Let $T = (V, E)$ be a directed tree with $n = |V|$ nodes, the $(n - 1) \times (n - 1)$ truncated matrix T_1 has the following properties:

1. it is possible to permute rows and columns of T_1 so that T_1 is a lower triangular matrix with ± 1 on its diagonal;

2. as a consequence of above T_1 is an invertible matrix;

3. as another consequence, $\det(T_1) = \pm 1$.

Corollary 4.3 The rank of the incidence matrix of a tree is $n - 1$ and the rank of the incidence matrix of a forest with k components is $n - k$.

Remark 4.4 In fact, the truncated matrix T_1 has the property that its inverse T_1^{-1} is al so a matrix of 0's and ± 1's.

Definition 4.2 Suppose $G = (V, E)$ is an undirected graph, with $|V| = n$. Suppose $E_1 \subseteq E$ is a subset of arcs with $|E_1| = n - 1$ such that the subgraph $T = (V,E_1)$ is a tree. Then T is called a *spanning tree* of G.

Remark 4.5 Only connected graphs have spanning tree. Also the concept of spanning trees extends to directed networks if we disregard the directions.

Lemma 4.6 Let $G = (V, E)$ be a *weakly connected* directed graph with incidence matrix A. Suppose we remove one node (one column of A), then a subset of $n - 1$ rows along with the remaining $n - 1$ columns of A forms a non-singular matrix if and only if the $n - 1$ rows correspond to the arcs of a *spanning tree*.

4.2 Network Optimization Models

Consider a distribution network, consisted of *supply nodes*, where a certain commodity is produced, *demand nodes*, where the commodity is required, and *transshipment nodes*, where there is no demand or supply: the commodity is simply passed through. The data of the problem are as follows:

1. Each supply node i has a maximum production capacity s_i;
2. Each demand node j has a minimum demand d_j;
3. each arc (i, j) has a cost c_{ij};
4. In some cases, there may be an upper bound or capacity on each arc (i, j), k_{ij} indicating the maximum amount that can pass.

The model consists of:

1. *Decision variable:* our goal is to find how much flow x_{ij} should we pass through each arc;
2. *Constraints:* we must make sure that the net amount of flow at each node should be consistent with its demand or supply. For a supply node i with a supply of s_i this implies:

   $$\sum_{j;(i,j)\in E} x_{ij} + s_i = \sum_{j;(j,i)\in E} x_{ji} \tag{19}$$

   For a demand node i with a demand d_i:

   $$\sum_{j;(i,j)\in E} x_{ij} = \sum_{j;(j,i)\in E} x_{ji} + d_i \tag{20}$$

   Finally, for a transshipment node i:

   $$\sum_{j;(i,j)\in E} x_{ij} = \sum_{j;(j,i)\in E} x_{ji} \tag{21}$$

   There is an easy way to combine all these. We assume every node is a demand node. Then a supply node is a demand node with a negative demand and a transshipment node is a demand node with a 0 demand. Then all these constraints can be expressed as:

   $$\sum_{j;(j,i)\in E} x_{ji} - \sum_{j;(i,j)\in E} x_{ij} = D_i \text{ for each node } i \tag{22}$$

   These are sometimes called *balance constraints.* If there are capacities on the arcs, then $x_{ij} \le K_{ij}$ for all i, j;
3. *Objective function:* The goal is to find a flow through each arc where the total cost is as small as possible: $\min \sum_{(i,j)\in E} c_{ij}x_{ij}$

Let's now state the entire problem at once

$$
\begin{aligned}
\min \quad & \sum_{ij} c_{ij} x_{ij} \\
\text{subject to:} \quad & \sum_{j;(j,i)\in E} x_{ji} - \sum_{j;(i,j)\in E} x_{ij} = D_i, \ \forall i, \\
& 0 \le x_{ij} \le K_{ij}, \ \forall (i,j).
\end{aligned} \tag{23}
$$

If we write the balance constraints of this problem in matrix form: $Ax = b$, then the matrix $A \in \mathbb{R}^{|V|\times|E|}$ is the transpose of the incidence matrix. This observation helps us characterize the dual and also the basic feasible solution (to be seen shortly). First, let us look at the dual: $A^\top y \le c$, i.e.,

$$y_j - y_i \le c_{ij} \text{ for each arc } (i,j) \tag{24}$$

and the objective function $\max \sum d_i y_i$, where d_i is the demand of node i. Note that if we work with the initial objective function then adding a constant amount to each y_i will also be feasible and optimal:

$$(y_j + a) - (y_i + a) \le c_{ij} \text{ and } \sum d_i(y_i + a) = \sum d_i y_i + a(\sum d_i) = \sum d_i y_i \tag{25}$$

So if we use the incidence matrix, then the dual will not have a unique solution. If we remove a column of $A^\top$ corresponding to a node i, then this means that in the dual we can set y_i to an arbitrary fixed constant, say 0. Let's discuss further the issue of linear dependence of columns of A.

The matrix A has linearly dependent rows, since in each column there is exactly one $+1$ and one -1. So if we add up all the rows, we will get the vector 0. Also note that for the *min-cost-flow* problem to be feasible, it is necessary that sum of supplies be equal to sum of demands. Let $\bar{A}$ be the matrix corresponding to removal of one of the rows of A. Based on the discussion above, we get the following useful fact:

Theorem 4.7 Let B be a basis corresponding to a basic solution, then columns of B correspond to a subset of $n-1$ arcs of the network forming a spanning tree. As a result all basic matrices B can be turned into a lower triangular matrix with ± 1 on its diagonal.

Remark 4.8 The statement above is true in particular for basic feasible solutions.

Suppose we have a feasible flow (basic feasible solution), roughly, the simplex method as applied to the minimum cost flow problem can be expressed as follows:

1. Find an initial basic feasible solution: in this case, it means find a set of arcs forming a spanning tree. All others arcs will have zero flow and using the easy triangular matrix B determine the flow in the spanning tree;

2. Find a non-basic arc whose inclusion into the basis could improve the cost. To do so, first calculate the dual variable y by solving $B^\top y = c_B$ when c_B consists of c_{ij} with (i,j) an arc of the spanning tree. This system is very easy: $y_j - y_i = c_{ij}$, starting from the fixed y_i, we can set other values backwards. Once all y_i are found, then for all arcs (i,j) not in the tree find reduced costs: $s_{ij} = c_{ij} - (y_j - y_i)$ and choose one where $s_{ij} < 0$ to enter the basis;

3. Adding (i, j) to the spanning tree will create a cycle. We go around the cycle and examine which arcs need to be reduced. Observe that the additional flow α along the new arc should be (i) added to the arcs in the cycle in the same direction as the new arc; and (ii) suffocated from the arcs in the opposite direction of the new arc. Since we have to keep everything larger than 0,

$$\alpha = \min_{\{(i,j) \text{ in cycle and } (i,j) \text{ opposite new arc}\}} x_{ij} \tag{26}$$

4. If all $s_{ij} \geq 0$, we can stop, the current flow x_{ij} is optimal.

The question now is how to find an initial basic feasible solution. Here is a way:

1. Add a new transshipment node v;

2. Draw new arcs from v to all demand nodes ($d_i > 0$) and cost $+1$;

3. Draw new arcs from supply nodes ($d_i < 0$) to v with cost $+1$ (include transshipment nodes with supply node);

4. Make the other arcs equal to 0.

The initial feasible flow is now clear: add s_i (supply amount on node i) to arc (i, v) if i is a supply node or transshipment node, add d_i to arc (v, i) if i is a demand node. Attempt to optimize this network, if at the optimal solution, optimal cost is 0, that is all new arcs have flow 0, we can remove the new node v, and the new arcs and the remaining flow is feasible for the original problem. If, however, the cost is bigger than 0 and some new links have positive flow, the original problem is infeasible.