ISyE 7203: Logistics and System Engineering

Rui Gong

August 26, 2025

Acknowledgements

These notes are based on the ISyE 7203 lectures given by Professor *Alejandro Toriello* in Fall 2025 at Georgia Institute of Technology.

Contents

1 Introduction

1.1 Network Flows

Definition 1.1

A network (graph) *is made of the following:*

- (N) *nodes (vertices): a finite set.*
- *For undirected network:* (E) *edges:* $\{i, j\}, i, j \in N$ *drawn as* $i - j$.
- *For directed network:* (A) *arcs* $(i, j), i, j \in N$ *drawn as* $i \to j$*, where we call* i *the tail and* j *the head of the arc.*

For this course, we focus on the directed networks and the following definitions are all on directed networks.

Definition 1.2

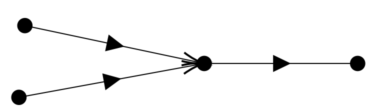
- path*: a sequence of arcs where the head of the first is the tail of the second and so forth, and no nodes repeat.*
- walk*: a sequence of arcs where the head of the first is the tail of the second and so forth, and nodes may repeat.*
- cycle*: a path except that the first and the last nodes are the same. If a cycles has repeated nodes in between, it is a* closed walk *instead.*

Definition 1.3: Connectedness

A directed network is:

- strongly connected*: for any* $i, j \in N$*, there exists an* $i - j$ *(directed) path and a* $j - i$ *directed path.*
- weakly connected*: for any* $i, j \in N$*, there exists an (undirected)* $i - j$ *path if we ignore orientations.*

Example 1.1. Consider the following network:



It is weakly but not strongly connected.

Definition 1.4: network flow

Given the network by (N, A), $b \in \mathbb{R}^N, c \in \mathbb{R}^A, u \in (\mathbb{R} \cup \{\infty\})^A$. b represents the net supply where i is a supplier, consumer, trans-shipment respectively if $b_i > 0, b_i < 0, b_i = 0$ respectively; c represents the cost, and u represent the capacity.
For modeling, X_{ij} is used to represent the amount of flow on $(i, j) \in A$. Then the network flow problem *is:*

$$\begin{aligned} \min \; & \sum_{a \in A} c_a x_a \\ \textit{s.t.} \; & \sum_{a \in \delta^+(i)} - \sum_{a \in \delta^-(i)} x_i = b_i, i \in N \\ & 0 \le x_a \le u_a, \; \forall a \in A \end{aligned} \tag{NF}$$

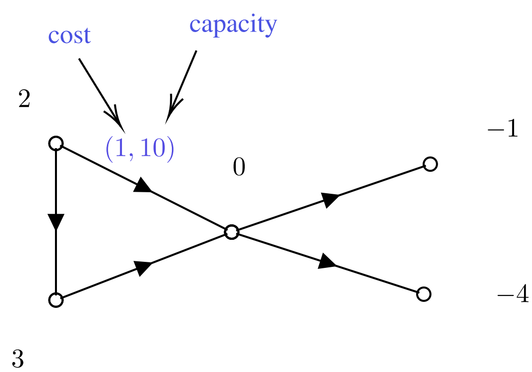
where $\delta^+(i)$ is the set of arcs having i as the tail and $\delta^-(i)$ is the set of arcs having i as the head. The equality constraints are called the flow balance constraint.

Remark. Aggregate the flow balance constraints, we get

$$\sum_i \left(\sum_{a \in \delta^+(i)} - \sum_{a \in \delta^-(i)} x_i \right) = \sum_{a \in A} x_a - x_a = 0 = \sum_i b_i.$$

Thus, $\sum_{i=1}^N b_i = 0$ is a needed assumption.

Example 1.2. The following is an instance of a network flow problem:



Theorem 1.5

The constraint matrix of (NF) *is always TU (totally unimodular). Thus, if b, u are integral, all extreme points are integral.*

Example 1.3 (Assignment Problem). Considering assigning n workers to n tasks, one for each. Let c_{ij} be the cost of assigning worker i to task j. In this way, we can consider $1, \ldots, n$ nodes as workers and $n+1, \ldots, 2n$ as tasks. For each node representing a worker, it has arcs from it to all task nodes. The supply of each worker and task are 1 and

-1 respetively. Then the problem can be written as:

$$\begin{aligned} \min\ & \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{s.t.}\ & \sum_{j=n+1}^{2n} x_{ij} = 1,\ i = 1,\dots,n \\ & \sum_{i=1}^{n} -x_{ij} = -1 \\ & 0 \le x_{ij}. \end{aligned}$$

Notice that there is no need to have $x_{ij} \le 1$, it is implied by the balance flow constraints and nonnegative constraints.

Remark. Indeed, the above example shows that the extreme points of the set of double stochastic matrices are permutation matrices.

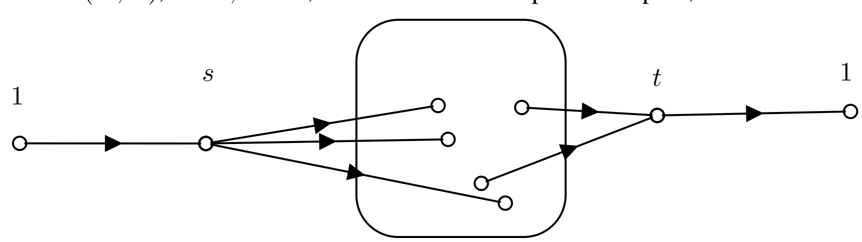
Example 1.4 (Transportation Problem). Given suppliers $i = 1,\dots,m$ with b_i units of supply and consumers $j = 1,\dots,n$ with d_j units of demand, let c_{ij} be the cost per unit from i to j. Assume that $\sum_i b_i = \sum_j d_j$, then we get the LP:

$$\begin{aligned} \min\ & \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{s.t.}\ & \sum_{j=n+1}^{2n} x_{ij} = b_i,\ i = 1,\dots,m \\ & \sum_{i=1}^{n} -x_{ij} = d_j,\ j = 1,\dots,n \\ & 0 \le x_{ij} \le u_{ij}. \end{aligned}$$

What if $\sum_i b_i > \sum_j d_j$?
We can consider adding a dummy customer whose demand is $\sum_i b_i - \sum_j d_j$. That is, we require a dummy customer takes all the extra supply.

1.2 Shortest Path

Given a directed network (N, A), and $s, t \in N$, $c \in \mathbb{R}^A$. Goal: cheapest $s - t$ path,

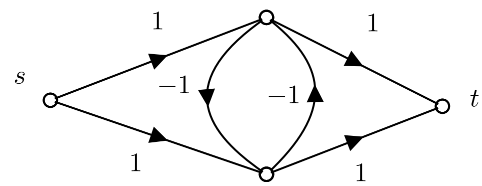


$$\begin{aligned} \min\ & \sum_{a\in A} c_a x_a \\ \text{s.t.}\ & \sum_{\delta^+(s)} x_a = 1 \\ & -\sum_{\delta^-(s)} x_a = -1 \\ & \sum_{\delta^+(s)} x_a - \sum_{\delta^-(s)} x_a = 0, \forall i \neq s, t. \end{aligned}$$

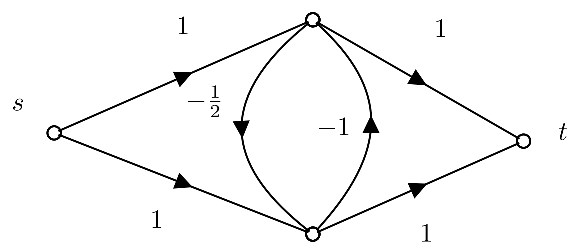
where

$$b_i = \begin{cases} 1, \ i = s \\ -1, \ i = t \\ 0, \ \text{o.w.} \end{cases}.$$

However, this formulation has a problem, consider a cycle in the network with all arcs having cost -1, then the LP is unbounded by repeating on the cycle.



Even if we add $x_a \leq 1$, we can still have a walk but not necessarily a path.



Notice that for the network above, the shortest path has cost 1, but the LP can give a shortest path with cost $1/2$ even if having constraints $x_a \leq 1$, which actually represents a walk that goes through the cycle once.

Condition: SP can be solved as a network flow problem if the network does not have directed cycles of net negative cost. e.g.: non-negative costs, no (directed) cycles.

Remark. In general, if there is a negative cost cycle, then the problem is NP-hard.

Theorem 1.6: Principle of Optimality

Assume that there is cycle of net negative cost. If P is an $s-t$ shortest path and $i, j \in P$ (i precedes j in P), then the $i-j$ sub-path $\hat{P}$ is an $i-j$ shortest path.

Proof. Suppose there is an $i-j$ path $\tilde{P}$ for contradiction such that $\sum_{a \in \tilde{P}} c_a < \sum_{a \in \hat{P}} c_a$. Replace $\hat{P}$ with $\tilde{P}$ in P to obtain P'. If P' is a path, we are done. If it is a walk, then there exists cycle which we can delete can strictly decrease the cost. That is, P' is a cheaper path than P, contradiction. □

Corollary 1.7

Let y_i^* be the shortest path cost from i to t. Then $y_i^* \leq c_{ij} + y_j^*, \ \forall (i,j) \in A$.
(Same thing in reverse: $\hat{y}_i$ is a shortest path cost from s to i, then $\hat{y}_j \leq c_{ij} + \hat{y}_i$.

Consider the dual of the LP:

$$\begin{aligned} &\max y_s - y_t \\ &\text{s.t. } y_i - y_j \leq c_{ij}, \ \forall (i,j) \in A \end{aligned}$$

Note. if (y_i) is feasible, then $(y_i + \alpha)$ is also feasible for any $\alpha \in \mathbb{R}$ and has the same objective value. Thus, the linearity space is no empty. Thus, there is at least one constraint in the primal that is redundant, and we can then assume the dual variable corresponding to the redundant primal constraint to be 0. If we sum all the primal constraint up, we can see that the constraint corresponding to t is redundant and thus WLOG, assume $y_t = 0$. Then we have,

$$\begin{aligned} &\max y_s \\ &\text{s.t. } y_i \le c_{ij} + y_j,\ \forall (i,j) \in A, \end{aligned}$$

which is we have from the principle of optimality corollary; opt y^* encodes distances to t. With complementarity, (i, j) can only be in a shortest path if $y_i^* = c_{ij} + y_j^*$.

Analogously, if we set $y_s = 0$ instead, then $-y$ encodes distances from s.

Definition 1.8: Directed Acyclic Graph

A directed graph is called a directed acyclic graph *if it does not contain a cycle.*

Definition 1.9: Topological Sort

A topological sort *of a directed graph is a labeling of nodes such that $i < j$ for every $(i, j) \in A$.*

Proposition 1.10

A network is a directed acyclic graph iff it has a topological sort.

Proof.

- ($\Longleftarrow$): If there is a cycle, then following the cycle, $i < j_1 < \cdots < j_k < i$, contradiction.
- ($\Longrightarrow$):

  Claim. If acyclic, there exists $v \in N$ with $\delta^-(v) = \emptyset$.

  Proof of the Claim. Perform a search starting from an arbitrary node backwards following the incoming arcs. By the finiteness of the network, we either get a cycle of a node without incoming arcs. □

  Input: (N, A), $i \leftarrow 1$.
  while $N \neq \emptyset$ **do**
      choose any $v \in N$ with $\delta^-(v) = \emptyset$
      $\text{label(v)} \leftarrow$ i; $i + +$
      $N \leftarrow N \setminus v$, update A
  end while

  Claim. At assignment of label i, $(1, \ldots, i)$ is a a valid top. sort for the network induced by $1, \ldots, i$.

  Proof of the Claim.

  - Base case: $i = 1$, trivial.
  - Induction: $i \geq 2$, $(1, \ldots, i - i)$ is top. sort, any arc with head i must have tail $\leq i - 1$.

  □

□

Algorithm 1 Bellman-Ford

$y_n \leftarrow 0$
for $i = n-1, n-2, \dots, 1$ **do**
  $y_i \leftarrow \min_{(i,j)\in A}\{c_{ij} + y_j\}$
  [succ(i) $\leftarrow$ j]
end for

1.3 Shortest Path on Direct Acyclic Networks

Given $c \in \mathbb{R}^A$, we can assume $\delta^-(s) = \delta^+(t) = \emptyset$. Find topological sort with $(s = 1, \dots, t = n)$.

- Running Time: $\mathcal{O}(m)$, topological sort and assigning values to y_i.
- Correctness: induction (backwards) at step i, labels $y_i, \dots, y_n$ are correct.

Proof.
- $i = n$, trivial.
- $i < n$, suppose $y_i = c_{ij} + y_j = c_{ij} + \sum_{a\in P} c_a$ where P is a shortest $j-t$ path. All $i-t$ paths use same arc (i,k) implies that $y_i \le c_{ik} + y_k \le$ cost of any $i-t$ path that uses (i,k). That is, the cost of any $i-t$ path uses (i,k) has cost more than y_i; thus, when we have y_i equal to the cost of $i-t$ path through (i,j) and shortest path from j to t, also, this value is the smallest among all $c_{ik} + y_k$ thus less than or equal to the cost of all paths from $i-t$ through (i,k).

□

Notice that, the reason that we need a top. sort is that for $(i,j) \in A$, i is the tail and j is the head, by the top. sort, j must be greater than i.

BF on General Networks use "stages" Paths have $\le n-1$ arcs. Then, the induction claim becomes:

$y_t^0 \leftarrow 0; y_i^0 \leftarrow \infty, \forall i \neq t$
for $k = 1, \dots, n-1$ **do**
  for $i \in N \setminus t$ **do**
    $y_i^k := \min\{y_i^{k-1}, \min_{(i,j)\in A}\{c_{ij} + y_j^{k-1}\}\}$
  end for
end for

Claim. y_i^k is the $i-t$ Shortest path cost when using $\le k$ arcs.

Proof. Similar to the proof of the previous induction claim. □

The running time now becomes $\mathcal{O}(mn)$ might be worse than Diksjtra when the costs are nonnegative.

Theorem 1.11: I

the network has a negative cost cycle, following the above algorithm, $y_i^n < y_i^{n-1}$ for some i.

Proof. Take a cycle C, and suppose that $y_i^n = y_i^{n-1}$ for all $i \in C$. Then,

$$y_i^n \le c_{ij} + y_j^{n-1} = c_{ij} + y_j^n \text{ for } (i,j) \in C.$$

Thus,

$$\sum_{(i,j)\in C} y_i^n - y_j^n \le c_{ij} \implies 0 \le \sum_{(i,j)\in C} c_{ij}.$$

□