ISyE 7203: Logistics and System Engineering

Rui Gong

September 28, 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

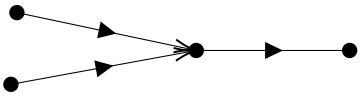
- path*: a sequence of arcs where the head of the first is the tail of the second and so forth, and no nodes repeat.*
- walk*: a sequence of arcs where the head of the first is the tail of the second and so forth, and nodes may repeat.*
- cycle*: a path except that the first and the last nodes are the 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:*

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

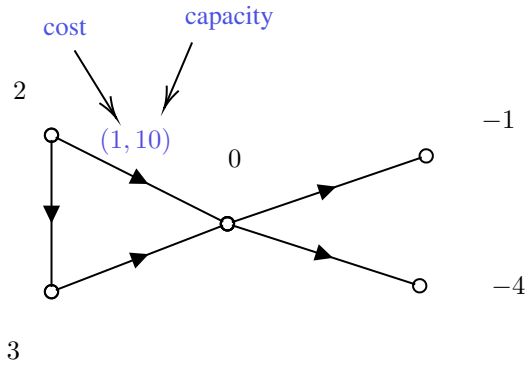
where $\delta^+(i)$ is the set of arcs having i as the tail and $\delta^-(i)$ is the set of arcs having i as the head. The equality constraints are called the flow balance constraint.

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 $1, \dots, n$ nodes as workers and $n+1, \dots, 2n$ 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.

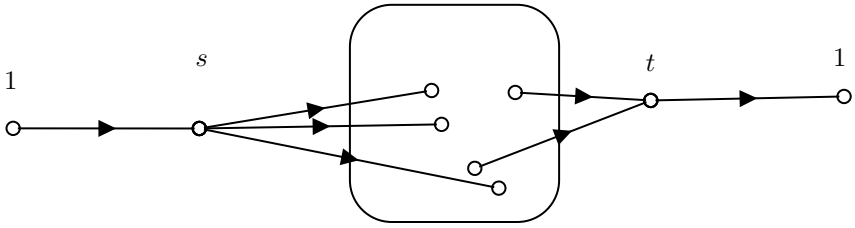
Example 1.4 (Transportation Problem). Given suppliers $i = 1,\dots,m$ with b_i units of supply and consumers $j = 1,\dots,n$ with d_j units of demand, let c_{ij} be the cost per unit from i to j. Assume that $\sum_i b_i = \sum_j d_j$, then we get the LP:

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

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,

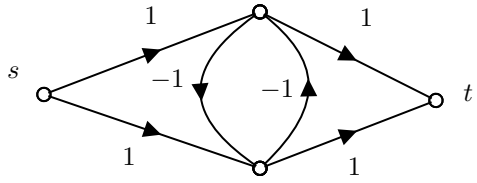


$$\begin{aligned}\min \ & \sum_{a\in A} c_a x_a \\ \text{s.t.} \ & \sum_{\delta^+(s)} x_a = 1 \\ & -\sum_{\delta^-(s)} x_a = -1 \\ & \sum_{\delta^+(s)} x_a - \sum_{\delta^-(s)} x_a = 0, \forall i \ne s,t.\end{aligned}$$

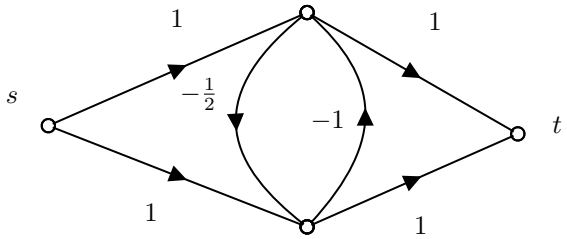
where

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

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,

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

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 $\delta^-(v) = \emptyset$
  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 $i = n-1, n-2, \ldots, 1$ **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 $(s = 1, \ldots, t = n)$.

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

$y_t^0 \leftarrow 0; y_i^0 \leftarrow \infty, \forall i \ne t$
for $k = 1, \ldots, n-1$ **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

If 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,

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

□

1.3.1 (Deterministic) Dynamic Programming (finite state, action, horizon)

Another name for shortest path on directed acyclic network, Knapsack:

$$\begin{aligned}\max\ & \sum_{i=1}^{n} c_i x_i \\ \text{s.t. } & \sum_{i=1}^{n} a_i x_i \le b \\ & a \in \mathbb{N}^n, b \in \mathbb{N}, x_i \in \{0,1\}, \forall i = 1,\dots,n.\end{aligned}$$

Knapsack as DP [SP]; nodes [states]; arcs[actions]. Let the current candidate item be $i = 1,\dots,n$, and the remaining capacity $s = 0,\dots,b$. Then, consider the sate (i,s) which has arcs from it to $(i+1,s)$ without any cost and to $(i+1, s-a_i)$ where $s \ge a_i$ with cost $-c_i$. Thus, the number of nodes is $\mathcal{O}(nb)$, the number of arcs is $\mathcal{O}(nb)$ and the start point is $(1,b)$. The BF complexity is $\mathcal{O}(nb)$ which, in precise, is $\mathcal{O}(n2^{\log_2 b})$

1.4 Max Flow

Given a network (N,A); source s and a sink t in N; arc capacities $v \in (\mathbb{R}_+ \cup \{\infty\})^A$. Add a dummy arc from t to s. How much can flow from s to t?

$$\begin{aligned}\min\ & -x_{ts} \\ \text{s.t. } & \sum_{\delta^+(s)} x_a + \sum_{\delta^-(t)} x_a = x_{ts} \\ & \sum_{\delta^+(i)} x_a - \sum_{\delta^-(i)} x_a = 0,\ \forall i \in N \setminus \{s,t\} \\ & 0 \le x_a \le u_a,\ \forall a \in \check{A} \\ & 0 = x_a, a \in \hat{A}\end{aligned}$$

Definition 1.12

s-t cut: $F \subseteq A$ such that $(N, A \setminus F)$ has no s-t path; capacity *of a cut F: $u(F) = \sum_{a\in F} u_a$;* minimal cuts: *$S \subseteq N \setminus t, S \ni s$, where we let $\delta_+(S)$ be the cut defined by S.*

Theorem 1.13

Weak Duality: for any feasible s-t flow x and any cut S,

$$x_{ts} \le u(S)$$

Proof. $x_{ts} + \sum_{\delta^+(S)} x_a = \sum_{\delta^+(S)} x_a - \sum_{\delta^-(S)} x_a \le \sum_{\delta^+(S)} u_a$. □

Theorem 1.14: [Max-Flow Min-Cut]

$$\max_{s-t\ flows} x_{ts} = \min_{s\in S\subseteq N\setminus t} u(S)$$

Dual of the LP:

$$\begin{aligned}\min_{a\in\check{A}} \; & u_a z_a \\ & y_t - y_s \geq 1 \\ & y_i - y_j \geq 0, \forall (i,j) \in \hat{A} \\ & y_i - y_j + z_{ij} \geq 0, \forall (i,j) \in \check{A} \\ & z \geq 0.\end{aligned}$$

Reminder of the primal:

$$\begin{aligned}\max_{x\geq 0} \; & x_{ts} \\ & \sum_{\delta^+(i)} x_a - \sum_{\delta^-(i)} x_a = 0, \; \forall i \in N \\ & x_a \leq u_a, \; \forall a \in \check{A}\end{aligned}$$

WLOG, set $y_s = 0$, then we can set $y_t = 1$. By totally unimodular, we can set $y \in \mathbb{Z}^N$ and $z_a \in Z_a$, and we need $y \in \{0,1\}^N$ which implies $Z_{ij} = 1$ when $y_j = 1$, $y_i = 0$ and 0 otherwise). Then, the optimal $y^* \in \{0,1\}^n$ defines the cut. $S = \{i : y_i^* = 0\}$, $z_{ij}^* = 1$ correspond to arcs in $\delta^+(S)$.

- $y \geq 0$: suppose $y_i < 0$ for some i. Replace all such y_i by 0.
- $y \leq 1$: set $y_i \leftarrow 1$ if $y_i > 1$.
- Complementary Slackness:

$$\begin{aligned}& x_{ij}(y_i - y_j) = 0, \; (i,j) \in \hat{A} \\ & x_{ij}(y_i - y_j + z_{ij}) = 0, \; (i,j) \in \check{A} \\ & z_{ij}(u_{ij} - x_{ij}) = 0, \; (i,j) \in \check{A}.\end{aligned}$$

  Thus, if $y_i > y_j$, then x_{ij}, so for this arc (i,j), there is no flow, which agrees with $x_{ts} = \sum_{\delta^+(S)} x_a - \sum_{\delta^-(S)} x_a \leq \sum_{\delta^+(S)} u_a$ requires $\sum_{\delta^-(S)} x_a$ for max flow being equal to the min cut. Similarly, $X_{ij} = 0$ only when $z_{ij} = 0$ and $y_i > y_j$, same to the above. $z_{ij} > 0$ if and only if $u_{ij} = x_{ij}$, so there is only flow when the arc is used at capacity.

1.4.1 Max Flow Applications

Theorem 1.15: Hall's Theorem

Consider a bipartite network (N, E) where $N = V \cup W$, $V \cap W = \emptyset$, $|V| = |W| = n$. Necessary and sufficient condition for perfect matching (marriage condition): for $X \subseteq V$, need $|\Gamma(X)| \geq |X|$.

Proof. Let s having arcs from s to each vertex in V, and each vertex has arcs to t, each of them has capacity 1. For arcs between V, W, let their capacity be ∞.

Clearly, (N, E) has a perfect matching iff the max flow in the network built has value $|V| = |W| = n$. If the max flow has value $< n$, we want to show that some $X \subseteq V$ violates the marriage condition. Suppose so, then the min cut has capacity $< n$ (by the fact any max flow has to use all capacity on the arcs). Then this cut cannot have any arc between V, W, so the cut S has $S \cap V \neq \emptyset$, and if $i \in S \cap V$, then $j \in \delta(i)$ also has $j \in S$. Capacity of S is $|V \setminus S| + |W \setminus S| = n - |V \cap S| + |\delta(S \cap V)| < n$ since S is a min cut. Then $|\Gamma(S \cap V)| < |S \cap V|$, contradiction. □

2 Traveling Salesman Problem

2.1 TSP

Consider a depot: 0; customers: $N = \{1, \dots, n\}$; c_{ij}: cost of direct travel from i to j. Goal: minimum cost Hamiltonian cycle "tour". Assume $c \geq 0$ and a complete network; triangle inequality: $c_{ij} \leq c_{ik} + c_{kj}$. Maybe: symmetry: $c_{ij} = c_{ji}$.

IP formulation:

$$x_{ij} = \begin{cases} 1, & \text{if tour includes } (i,j); \\ 0, & \text{otherwise.} \end{cases}$$

$$\begin{aligned} \min \quad & \sum_{i,j \in N \cup \{0\}, i \neq j} c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{\delta^+(i)} x_a = \sum_{\delta^-(i)} x_a = 1, \ \forall i \in N \cup \{0\} \\ & \sum_{\delta^+(S)} x_a \geq 1, \ \emptyset \neq S \subseteq N \\ & x_{ij} \in \{0,1\}. \end{aligned}$$

where the second constraint is called the subtour elimination contraint, there would be exponentially many of such constraints. For undirected graphs, we have $\sum_{\delta(i)} x_e = 2$ and $\sum_{\delta(S)} x_e \geq 2$ instead.

DP formulation (Held+Karp): When building a tour, need current location and remaining customers to visit.

- State: $(i, S), i \in N \cup \{0\}, S \subseteq N$.
- Action: $j \in S$ if $S \neq \emptyset$, 0 if $S = \emptyset$.
- Number of states: $\Theta(n2^n)$.
- Number of actions: $\Theta(n^2 2^n)$.

Lower Bounds and Relaxations Forward Star: cheapest arc from every forward start

$$\begin{aligned} \min_{x \geq 0} \ & \sum c_a x_a \\ & \sum_{\delta^+(i)} x_a = 1, \ i \in N \cup \{0\}. \end{aligned}$$

Assignment Bound

$$\begin{aligned} \min_{x \geq 0} \ & \sum c_a x_a \\ & \sum_{\delta^+(i)} x_a = 1 \\ & \sum_{\delta^-(i)} x_a = 1. \end{aligned}$$

Min. Spanning Tree (for the undirected graph) Hamiltonian cycle is a Hamiltonian path plus an edge. Hamiltonian path is a spanning tree. If $c \geq 0$, MST is a lower bound.

Take $S \subseteq N$,

$$\begin{aligned}&\sum_{i\in S}\sum_{\delta(i)} x_e = 2|S| \\ =&2\sum_{e\subseteq S} x_e + \sum_{\delta(S)} x_e\end{aligned}$$

By subtracting $\sum_{\delta(S)} x_e \geq 2$, we get

$$\sum_{e\subseteq S} x_e \leq |S|-1.$$

Then, $\sum_{i\in N\cup\{0\}}\sum_{\delta(i)} x_e = 2(n+1) = 2\sum_e x_e$. Consider

$$\begin{aligned}\min_{x\geq 0}\ &\sum_e c_e x_e \\ &\sum_e x_e = n+1 \\ &\sum_{e\subseteq S} x_e \leq |S|-1, S \subsetneq N\cup\{0\}\end{aligned}$$

Notice that when we change $n+1$ to n, the corresponding coefficient in the objective function of the dual decrease. Since the inequalities are $\leq$, the corresponding variables in the dual are ≤ 0, the only possible nonnegative variable is the one corresponding to the equality. However, since the objective function of the primal is nonnegative (when $c \geq 0$), the dual variable corresponding to the equality has to be nonnegative, changing $n+1$ to n decreases the objective value of the dual, thus it's a lower bound.

0-tree a TSP tour is a Hamiltonian cycle through N plus two edges connecting N to O (so you create a unique cycle including 0). Can be optimized with greedy algorithm (like minimum spanning tree problem).

$$\begin{aligned}\min\ &\sum_e c_e x_e \\ &\sum_{\delta(0)} x_e = 2 \\ &\sum_{e\subseteq N} x_e = n-1 \\ &\sum_{e\subseteq S} x_e \leq |S|-1,\ \forall S \subseteq N \\ &0 \leq x_e \leq 1.\end{aligned}$$

0-Tree Bound + Lagrangian Relaxation for Symmetric TSP $\pi(2-\sum_{\delta(i)} x_e) = 0$ for any TSP tour, set $\pi_0 = 0$. Then we can consider the objective function

$$\sum_{i,j} c_{ij}x_{ij} + \sum_{i\in N\cup\{0\}} \pi_i(2-\sum_{\delta(i)} x_i) = \sum_{i,j}(c_{ij}-\pi_i-\pi_j)x_{ij} + 2\sum_i \pi_i,$$

which implies that for any $\pi \in \mathbb{R}^N$, the cost of optimal 0-tree w.r.t. this objective plus $2\sum_i \pi_i$ gives a lower bound.

$$f(\pi) = 2\sum_i \pi_i + \min_{\text{0-tree}T}\{\sum_{ij\in T}(c_{ij}-\pi_i-\pi_j)\},$$

want $\max_{\pi\in\mathbb{R}^N} f(\pi)$. Notice that f is concave and piece-wise linear in π because $\sum_{ij\in T}(c_{ij}-\pi_i-\pi_j)$ is linear in π.
Optimize 0-tree: $\deg(i)=1$, then increase π_i; $\deg(i)>2$, then decrease π_i.

$$\begin{aligned}\max_{\pi\in\mathbb{R}^N}\ & 2\sum_{i\in\mathbb{N}}\pi_i+\min_{x\geq 0}\sum_{i,j}(c_{ij}-\pi_i-\pi_j)x_{ij}\\ \text{s.t.}\ & \sum_{\delta(0)}x_e=2\\ & \sum_{e\subseteq N}x_e=n-1\\ & \sum_{e\subseteq S}x_e\leq |S|-1,\ \forall\emptyset\neq S\subseteq N\\ & [x_e\in\{0,1\}]\end{aligned}$$

where the constraint $\delta_{\delta(i)}x_e=2$ is dropped and we penalize it in the objective function. IP theory implies that, since the polyhedron defining this relaxation is integral, the value of the Lagrangian relaxation equals the LP relaxation.

$$\begin{aligned}\min\ & \sum_{i,j\in N\cup\{0\},i\neq j}c_{ij}x_{ij}\\ \text{s.t.}\ & \sum_{\delta^+(i)}x_a=\sum_{\delta^-(i)}x_a=1,\ \forall i\in N\cup\{0\}\\ & \sum_{\delta^+(S)}x_a\geq 1,\ \emptyset\neq S\subseteq N\ (\star)\end{aligned}$$

Let $\mathcal{S}\subseteq 2^N$.

---

while do:
  Solve LP with $(\star)$ for $S\in\mathcal{S}$, get solution $\hat{x}$.
  check: $\min_{\emptyset\neq S\subseteq N}\left\{\sum_{\delta^+(S)}\hat{x}_a\right\}<1$ (global) MINCUT
  if ≥ 1 **then**
    Done, exist
  else
    add minimizing S to $\mathcal{S}$
  end if
end while

2.2 Approximate LP

Relaxation technique for DP TSP: state (i,S)

$$\begin{aligned}\max\ & y_{0,N}\\ \text{s.t.}\ & y_{i,S}\leq c_{ij}+y_{i,S\setminus j},\ \forall i\in N,S\subseteq N\setminus i,S\neq\emptyset,\\ & y_{i,\emptyset}\leq c_{i0},\ \forall i\in N,\end{aligned}$$

any feasible y gives a lower bound.

A primal relaxation is equivalent to a dual restriction.

$$y_{i,S}\approx\lambda\cdot b_{i,S}$$

where $b_{i,S}$ is a basis vector and λ are weights. e.g. $y_{i,S}\approx\lambda|S|+\sum_{k\in N\cup 0}\lambda_k\mathbb{1}(i=k)$.

Example 2.1. $y_{i,S} = \sum_{k\in S\cup i}\lambda_k + \sum_{k\in S\cup 0}\mu_k,\ \lambda,\mu\in\mathbb{R}^{N\cup 0}$. Interpret λ as the cost of leaving the node and μ as the cost of entering the node.

$$y_{i,S} - y_{j,S\setminus j} = \lambda_i + \mu_j.$$

$$\begin{aligned}\max\ & y_{0,N} = \sum_{k\in N\cup 0}(\lambda_k+\mu_k)\\ \text{s.t. }& \lambda_i+\mu_j\le c_{ij},\ i,j\in N\cup 0.\end{aligned}$$

And the dual is

$$\begin{aligned}\min\ & \sum_{i,j} c_{ij}x_{ij}\\ \text{s.t. }& \sum_{\delta^+(i)} x_a + \sum_{\delta^-(i)} x_a = 1,\ \forall i\end{aligned}$$

2.3 TSP Heuristic

We consider the worst-case analysi ofa minimization problem.

Definition 2.1

Let $C_H(I)$ be the cost of a heuristic H for an instance I, and $C^*(I)$ be the optimal cost of I. If $C_H(I)\le \alpha C^*(I)$ for all I, then we say H is an α-approximation algorithm *where $\alpha\in[1,\infty)$.*

2.3.1 Minimum Spanning Tree(MST) Heuristic for Symmetric TSP

Assume triangle inequality is satisfied.

Algorithm 2 MST

Step 1: Construct MST on $N\cup\{0\}$. Call it T.
Step 2: Replace each edge with two arcs in T with the same weight. The created graph has vertices with all degrees even. That is, if $\{u,v\}\in T$, delete it and add arcs $(u,v),(v,u)$.
Step 3: Construct Eulerian tour through T, which is guaranteed to exists by the even degree property.
Step 4: Every time the tour repeats a node, "short cut" to the next node. That is, in the tour, delete the repeated node and keep going, this only improves the tour by triangle inequality. Output this Hamiltonian cycle.

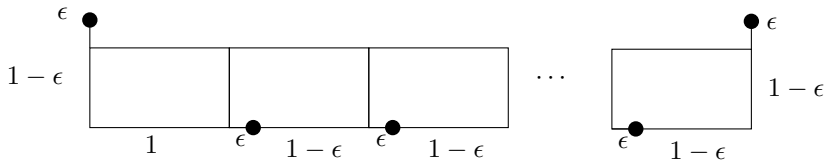
Theorem 2.2

$$C_{Alg2}(I)\le 2C^*(I)$$

for all symmetric instances with triangle inequality.

Proof. $C_{MST}\le C_{TSP}$ by the fact that any TSP can become a MST by deleting an edge. $C_{Alg2}\le 2C_{MST}$ by changing the edges to arcs and triangle inequality. □

Example 2.2 (Tightness of the above Theorem (Johnson and Papadimitriou 1985)). Consider a graph of $(1-\epsilon)\times 1$ rectangles repeating n times and connect by the $(1-\epsilon)$ edge one by one. The set of vertices are $\{(0,0),(1,0),(1+\epsilon,0),(2,0),\dots,(n-1+\epsilon,0),(n,0),(0,1-\epsilon),(1,1-\epsilon),\dots,(n,1-\epsilon),(0,1),(n,1)\}$. Use the Manhatten distance for triangles. Then MST has cost $n(2-\epsilon)+(1-\epsilon)+2\epsilon$, MST heuristic has $n(3-2\epsilon)+2\epsilon+(2\epsilon+n)$, OPT has $2n+2+4\epsilon$. The ratio of $C_{Alg2}/C_{\text{OPT}}=2$ as $n\to\infty$.



2.3.2 Greedy Heuristics

Algorithm 3 Nearest neighbor

$i \leftarrow 0, S \leftarrow N, T \leftarrow \emptyset$.
while $S \neq \emptyset$ **do**
  $j \leftarrow \arg\min_{k \in S} c_{ik}$
  $T \leftarrow T \cup \{ij\}, S \leftarrow S \setminus j$
  $i \leftarrow j$
end while
$T \leftarrow T \cup \{i0\}$

To make it better: *insertion*.

Algorithm 4 cheapest insertion

$j \leftarrow \arg\min_{k \in N}\{c_{0k}\}$
$T \leftarrow \{\{0, j\}, \{j, 0\}\}, S \leftarrow N \setminus j$
while $S \neq \emptyset$ **do**
  choose $j \in S$ insert into T as cheaply as possible: $\min c_{ij} + c_{jk} - c_{ik}, \forall \{i, k\} \in T$
  $T \leftarrow T \cup \{\{i, j\}, \{i, k\}\} \setminus \{i, k\}$
  $S \leftarrow S \setminus j$.
end while

Theorem 2.3

$CI(I) \leq 2\,\mathrm{OPT}(I)$ for symmetric I with $c \geq 0$, triangle inequality. The ratio is tight.

Proof. Let Ξ be a minimum spanning tree. We inductively maintain it as a forest, s.t. each component contains one node of T (the tour) and vice versa. Initially, $0 \in \Xi$ and Ξ is a tree. Induction: we insert j, some component Ξ_j contains j and also $\ell \in T$. In the unique $j - \ell$ path in Ξ_j, delete edges $\hat{e}$ incident to ℓ. Proof follows if $c_{ij} + c_{jk} - c_{ik} \leq 2c_{\hat{e}}$. Denote $\hat{e} = \{\ell, m\}$. Consider a neighbor ℓ' of ℓ in the tour T, then

$$c_{ij} + c_{jk} - c_{ik} \leq c_{\ell m} + c_{\ell' m} - c_{\ell\ell'} \leq 2c_{\hat{\ell m}} = 2c_{\hat{e}}.$$

Apply this inequality n times to get the result.
Initial cost: $c_{0i} + c_{i0}$. Insertion at k step: $\mathrm{cost}(T_k) - \mathrm{cost}(T_{k-1})$. The proof also holds for "nearest" insertion, where you only consider the nearest vertices to tour as candidates for j; in other words, you only consider the vertices like m in the above proof. □

Example 2.3. The above approximation factor is also tight. Consider a cycle with n vertices and a depot 0 and for $i \neq j \in [n]$, the c_{ij} is defined as the length of the shortest path between i, j. Consider a tour $0, n, n-1, 0$, and if I insert $n-2$, I can delete the edge $n, n-1$ or $n-1, 0$, both of them give cost increase of 2. If we keep deleting edge $n, n-1$, we have cost $2n$ but the opt is $n+1$.

2.3.3 Christofides' Heuristic

Definition 2.4: Eulerian Network

A network is Eulerian if it is connected and the degrees of all vertices are even.

For MST heuristics, we do not need to duplicate edges but only even degrees.

$$\sum_{i \in N} \deg(i) = 2|E|.$$

Thus, the number of vertices with odd degrees is even.

Compute an MST, T of $N \cup \{0\}$.
find minimum cost perfect matching of odd-degree nodes in T, M. Then $T \cup M$ is Eulerian.
Proceed as in MST heuristic.

Theorem 2.5

$Ch(I) \leq \frac{3}{2} \operatorname{OPT}(I)$ *for symmetric* I *with* $c \geq 0$*, triangle inequality.*

Proof. Consider the optimal tour, short-cut it so only odd-degree nodes in the MST remain, which has not larger cost by triangle inequality. Then, since there are even number of such nodes, the resulting tour is a cycle which can be split into two perfect matchings. Any of these matchings has cost greater than or equal to the cost M, thus

$$\operatorname{OPT} = c(C_{N \cup \{0\}}) \geq c(C_{odd}) = c(M_1) + c(M_2)$$

and

$$c(M) \leq \min\{c(M_1), c(M_2)\} \implies c(M) \frac{1}{2} \operatorname{OPT}.$$

□

Example 2.4. Christofides' Heuristic is tight.

2.4 Integrality Gap

how good is a bound in the worst case?

$$\alpha \operatorname{OPT}(I) \leq LB(I) \leq \operatorname{OPT}(I),$$

for $\alpha \in (0, 1]$.

Theorem 2.6

The LP relaxation of the subtour formulation is at worst $2/3$ *of optimum for symmetric instances with* $c \geq 0$*, triangle inequality.*

$$\begin{aligned} \min_{x \geq 0} \quad & \sum_e c_e x_e \\ \text{s.t.} \quad & \sum_{\delta(i)} x_e = 2, \ \forall i \in N \cup 0 \\ & \sum_{\delta(s)} x_e \geq 2, \ \forall \emptyset \neq S \subseteq N \end{aligned}$$

Proof sketch. We showed that $MST \leq LP$. One can similarly show that $c(M) \leq \frac{1}{2} LP$. □

2.5 Improvement Heuristics

Local search/neighborhood search. Single step above takes $O(n^2)$ time.

Algorithm 5 2-opt

Tour T
for non-consecutive edges $\{i,j\},\{k,\ell\} \in T$ **do**
  check $c_{ik} + c_{j\ell} < c_{ij} + c_{k\ell}$
  if true, replace in T.
end for

2.6 Metaheuristics

framework to generate heuristics solutions.
heuristic: procedure that generates (hopefully) a solution for an optimization problem, where the solution is (hopefully) feasible of high quality, and the procedure (hopefully) runs efficiently.

- NS
- Genetic algorithm
- simulated annealing
- Tabu

2.7 Approximate DP

$$y^*_{i,S} = \min_{j \in S}\{c_{ij} + y^*_{j,S\setminus j}\}.$$

where the optimal solution should be

$$\arg\min_{j \in S}\{c_{ij} + y^*_{j,S\setminus j}\}.$$

Instead of computing the "real" value function, we replace it with an estimation $\tilde{y}$. For example,

- "rollout": $\tilde{y}_{i,S} \geq y^*_{i,S}$
- "price-directed": $\tilde{y}_{i,S} \leq y^*_{i,S}$

Rollout $y^{NN}_{i,S}$: cost of nearest neighbor starting from i through S ending at 0, which is $c_{ij} + y^{NN}_{j,S\setminus j}$.
Nearest neighbor takes $O(n^2)$ so single step takes $O(n^3)$ time so $O(n^4)$ in total.
Rollout can be "sequentially improving".

Price-Directed: using approximate LP. ex:

$$y^A_{i,S} = \sum_{j \in S \cup i} \lambda_j + \sum_{j \in S \cup 0} \mu_j$$

at (i,S),

$$\min_{j \in S}\{c_{ij} + \sum_{k \in S} \lambda_k + \sum_{k \in S\setminus j \cup -} \mu_k\}.$$

which is equivalent to

$$\min_{j \in S}\{c_{ij} - \mu_j\}.$$

3 Vehicle Routing

Comparing to TSP: multiple routes instead of one. ex: capacity, duration limit, time windows.
Most basic: uniform demand + vehicle capacity $Q \in \mathbb{Z}$,

$$\begin{aligned} \min \ & \sum_{i,j} c_{ij} x_{ij} \\ & \sum_{\delta(i)} x_e = 2, \ i \in N \\ & \sum_{\delta(S)} x_e \geq 2, \ \emptyset \neq S \subseteq N. \end{aligned}$$

With the capacity, consider:

$$\begin{aligned} \min \ & \sum_{i,j} c_{ij} x_{ij} \\ & \sum_{\delta(i)} x_e = 2, \ i \in N \\ & \sum_{\delta(S)} x_e \geq 2 \left\lceil \frac{|S|}{Q} \right\rceil, \ \emptyset \neq S \subseteq N. \end{aligned}$$

3.1 route-then-cluster→ partition heuristic

Input: TSP tour $T = (0, 1, 2, \ldots, 0)$
for $q = 1, \ldots, Q$ **do**
  Evaluate $\{(1, \ldots, q), (q+1, \ldots, q+Q), \ldots\}$
end for
Return the best solution

Complexity: $O(Q \times (n/Q)) = O(n)$.

Lemma 3.1

$c_{VRP} \geq c_{TSP} \geq \frac{1}{Q} \sum_{i \in N} 2c_{oi}$.

Proof. Top bound follows from triangle inequality and shortcutting. Second: take an optimal partition of $N = N_1 \cup \ldots \cup N_k$,

$$c_{vrp} = \sum_{\ell=1}^{k} c_{TSP}(N_\ell) \geq \sum_{\ell} \max_{j \in N_\ell} \{2c_{0j}\} \geq \sum_{\ell} \frac{1}{|N_\ell|} \sum_{j \in N_\ell} 2c_{oj} \geq \frac{1}{Q} \sum_{j \in N} 2c_{oj}.$$

□

Theorem 3.2

Assume $c \geq 0$ and triangle inequality, symmetry. Suppose $c(T) \leq \alpha \times c_{TSP}$. Then the partition heuristic has $c_{PH} \leq c_{VRP}(1 + \alpha(1 - 1/Q))$.