CO 351 Review Questions (Part I)

David Duan

2019 Fall

Contents

1 Transshipment Problem

1.1 Transshipment Problem

1.1.1 Describe the transshipment problem.

1.1.2 What are the primal and dual LP for TP (explicit equations, no matrix)?

1.1.3 Define the incidence matrix and express LPs using the incidence matrix.

1.1.4 What's the complementary slackness conditions for TP?

1.2 Cycle, Spanning Tree, and Basis

1.2.1 Prove: The columns of \$M\$ corresponding to a cycle are linearly dependent.

1.2.2

Prove: If B is a subset of the arcs whose corresponding columns in M are linearly dependent, then B contains a cycle. (Hint: Show every vertex of the graph has $\deg \geq 2$.)

1.2.3 Combine (5) and (6): Describe a basis for the incidence matrix.

1.3 Network Simplex for TP

1.3.1 Define: node potential, reduced cost, feasible dual potential.

1.3.2 Prove: A feasible TP is unbounded iff there exists a negative dicycle.

1.3.3 Describe the network simplex algorithm for TP.

1.3.4 Describe the auxiliary TP and initialization procedure for network simplex.

1.4 Characterization of Infeasibility

1.4.1

Prove: A TP with digraph $D = (N, A)$ and node demand $b \in \mathbb{R}^N$ is infeasible if and only if there exists $S \subseteq N$ such that $b(S) < 0$ and $\delta(S) = \varnothing$. [Or equivalently, $b(S) > 0$ and $\delta(\bar{S}) = \varnothing$].

2 Minimum Cost Flow Problem

2.1 Minimum Cost Flow Problem

2.1.1 Describe the minimum cost flow problem.

2.1.2 What are the primal and dual LP for TP (explicit equations, no matrix)?

2.1.3 Describe the constraint matrix for MCFP.

2.1.4

Prove: A tree flow in MCFP consists of a spanning tree T and a feasible flow x where all arcs not in T satisfies $x_e \in \{0, c_e\}$.

2.1.5 What's the complementary slackness condition for MCFP?

2.1.6 Describe the network simplex for MCFP.

2.1.7

Prove: An MCFP is infeasible if and only if there exists $S \subseteq N$ such that $b(S) > c(\delta(\bar S))$ or $b(S) < -c(\delta(S))$.

2.2 MCFP Applications

2.2.1 Describe how you could solve the minimum cost perfect matching problem using MCFP.

2.2.2

Prove: If a MCFP has an optimal solution, and all capacities and node demands are integers, then there exists an integral optimal solution.

2.2.3 Justify your solution, that is, a solution of the MCFP instance is an optimal matching.

2.2.4

Describe how you could solve the airline scheduling problem using MCFP. (More importantly, you should be able to draw a diagram with $n=4$ cities.)

2.2.5

Describe how you could solve the catering problem using MCFP. (More importantly, you should be able to draw a diagram with $n=5$ days.)

2.2.6 Describe how you could transform a matrix with consecutive ones into an incidence matrix.

3 Shortest Dipath Problem

3.1 Shortest Dipath Problem

3.1.1 Describe the shortest dipath problem.

3.1.2 What are the primal and dual LP for shortest dipath problem?

3.1.3 Define: a characteristic vector of a path.

3.1.4

Prove: If $\bar x$ is an integral feasible solution to LP, then $\bar x$ is a sum of the characteristic vector of an s,t-dipath and a collection of dicycles.

3.1.5

Prove: If there are no negative dicycles, then our LP formulation has an optimal solution that is the characteristic vector of an s,t-dipath.

3.2 Ford's Algorithm

3.2.1

Prove: Let $D = (N,A)$ be a digraph with arc cost $w \in \R^A$ with no negative dicycle. If $v_1, \ldots, v_k$ is a shortest v_1, v_k-dipath, then $v_1, \ldots, v_i$ is a shortest v_1,v_i-dipath.

3.2.2 Define: A rooted tree at s.

3.2.3 What does Ford's algorithm accomplish?

3.2.4 Describe Ford's algorithm.

3.3 Correctness and Termination of Ford's Algorithm

3.3.1 Define: Predecessor graph.

3.3.2 Prove: Throughout the algorithm, $\bar w_e \leq 0$ for all arcs $e \in A(D_p)$.

3.3.3

Prove: Let $D = (N,A)$ be a digraph with weights $w \in \R^A$ and feasible potentials $y \in \R^N$. Let Q be an s,t-diwalk. Then $w(Q) \geq y_t - y_s$. Moreover, $w(Q) = y_t - y_s$ iff every arc of Q is an equality arc.

3.3.4

Prove: Let $D = (N,A)$ be a digraph with weight $w \in \R^A$. If D has a negative dicycle then D has no feasible potentials.

3.3.5

Prove: If D_p contains a dicycle (at any point in the algorithm), then D contains a negative dicycle and the algorithm does not terminate.

3.3.6

Prove: Suppose the algorithm terminates. Then D_p is a spanning tree of shortest dipaths rooted at s. Furthermore, y_v is the cost of a shortest s,v-dipath.

3.4 Bellman-Ford's Algorithm

3.4.1 Describe Bellman-Ford's algorithm.

3.4.2

Prove: Let d_v denote the *cost* of a shortest s,v-dipath. Suppose D does not have any negative dicycle. Then at any point in the algorithm $y_v \geq d_v$.

3.4.3

Prove: Suppose no negative dicycles exist. After the ith iteration, if there is a shortest s,v-dipath using at most i arcs, then $y_v = d_v$.

3.4.4

Prove: At the end of Bellman-Ford, if y is feasible, then $y_v = d_v$ for all $v \in N$. Otherwise, you can conclude there exists a negative dicycle.

3.5 Dijkstra's Algorithm

3.5.1 Describe the intuition of Dijkstra's algorithm.

3.5.2 Describe Dijkstra's algorithm.

3.5.3 Prove its correctness.

3.6 Application of Shortest Dipath

3.6.1 Describe how we could solve network reliability using shortest dipath.

3.6.2 Describe how we could solve currency exchange problem using shortest dipath.

1 Transshipment Problem

1.1 Transshipment Problem

1.1.1 Describe the transshipment problem.

Given a digraph $D = (N, A)$, node demands $b \in \mathbb{R}^N$, arc costs $w \in \mathbb{R}^A$, we want to find a feasible flow $x \in \mathbb{R}^A$ with minimum cost $w^T x$.

1.1.2 What are the primal and dual LP for TP (explicit equations, no matrix)?

The primal LP is given by

$$\begin{aligned} \min \quad & w^T x \\ s.t. \quad & x(\delta(\bar{v})) - x(\delta(v)) = b(v) \quad \forall v \in N \\ & x \geq \mathbf{0} \end{aligned}$$

The dual LP is given by

$$\begin{aligned} \max \quad & b^T y \\ s.t. \quad & y_v - y_u \leq w_{uv} \quad \forall uv \in A \end{aligned}$$

Equivalently, the dual constraint is $\bar{w}_{uv} = w_{uv} + y_u - y_v \geq 0$.

1.1.3 Define the incidence matrix and express LPs using the incidence matrix.

The primal LP can be written as

$$\begin{aligned} \min \quad & w^T x \\ s.t. \quad & Mx = b_v \quad (\forall v \in N) \\ & x \geq \mathbf{0} \\ \text{where} \\ & M \in \{-1, 0, 1\}^{|N| \times |A|} \\ & M_{v,ij} = \begin{cases} -1 & i = v, \text{ i.e., the arc leaves } v \\ 1 & j = v, \text{ i.e., the arc goes into } v \\ 0 & \text{else} \end{cases} \end{aligned}$$

The dual LP is thus

$$\begin{aligned} \max \quad & b^T y \\ s.t. \quad & M^T y \leq w \end{aligned}$$

1.1.4 What's the complementary slackness conditions for TP?

For each $uv \in A$, either $x_{uv} = 0$ or $y_v - y_u = w_{uv}$. Intuitively, for an arc $uv \in A$, either we don't use it at all, or we use it to its full potential.

1.2 Cycle, Spanning Tree, and Basis

1.2.1 Prove: The columns of M corresponding to a cycle are linearly dependent.

Proof. Let C be a cycle and M be its incidence matrix. Let M' be obtained from M by multiplying the columns corresponding to backward arcs in C by -1. Then M' is the incidence matrix of a dicycle C'. For each $v \in V(C)$, $\deg(v) = \deg(\bar{v}) = 1$, so there is exactly one entry of 1 and one entry of -1 in each row of M'. Then the sum of the entries of each row is 0 and thus the sum of columns of M' is zero. Since this is a non-trivial linear combination of columns of M with zero sum, M is linearly dependent. □

1.2.2 Prove: If B is a subset of the arcs whose corresponding columns in M are linearly dependent, then B contains a cycle. (Hint: Show every vertex of the graph has $\deg \geq 2$.)

Proof. Since the columns of B in M are linearly dependent, there exists a non-trivial linear combination of these columns that equal to zero. Let $B' := \{e_1, \dots, e_k\}$ be the subset of B whose columns receive non-zero coefficients in the linear combination, i.e., there exist non-zero c_i's where $c_1 M_{e_1} + \cdots + c_k M_{e_k} = 0$. Then for each node v, $c_1 M_{v,e_1} + \cdots c_k M_{v,e_k} = 0$, where M_{v,e_i} represents the entry for v in the column for e_i.

Let S be the subset of nodes v where $M_{v,e_i} \neq 0$ for at least one i. For a node $v \in S$, if $M_{v,e_i} \neq 0$, then $c_i M_{v,e_i} \neq 0$. Hence for each $v \in S$, there exist at least 2 arcs in B' whose columns in M have non-zero entries for v; each of these arcs is incident with v. It follows that the subgraph with S as the nodes and B' as the arcs have the property that every node has $\deg \geq 2$ and thus contains a cycle. □

1.2.3 Combine (5) and (6): Describe a basis for the incidence matrix.

Let M be the incidence matrix. A set of $|N| - 1$ columns of M is a basis iff the corresponding $|N| - 1$ arcs make up a spanning tree. Therefore, $\text{rank}(M) = |N| - 1$.

1.3 Network Simplex for TP

1.3.1 Define: node potential, reduced cost, feasible dual potential.

- Node potential: the dual solution $y \in \mathbb{R}^N$.
- Reduced cost: $\bar{w}_{uv} := w_{uv} + y_u - y_v$.
- Feasible dual potential: y is feasible if $\bar{w} \geq 0$.

1.3.2 Prove: A feasible TP is unbounded iff there exists a negative dicycle.

$\Longleftarrow$: Let C be a negative dicycle and x^* be a feasible flow. Define x^C where $x^C_e = t$ for $e \in C$ and 0 otherwise for $t \geq 0$. Observe $x^C(\delta(\bar{v})) = x^C(\delta(v)) = 0$ for each $v \in N$. Since C is a dicycle, and x is a feasible flow, $x^* + x^C$ is a feasible flow:

$$(x^* + x^C)(\delta(\bar{v})) - ((x^* + x^C)(\delta(v)) = [x^*(\delta(\bar{v})) - x^*(\delta(v))] + [x^C(\delta(\bar{v})) - x^C(\delta(v))] = b_v + 0 = b_v.$$

The objective value is $w^T(x^* + x^C) = w^T x^* + w^T x^C = w^T x^* + t \cdot w(C)$ where $w^T x^*$ is a constant. It follows that $w^T(x^* + x^C) \to -\infty$ as $t \to \infty$ as $w(C) < 0$.

$\Longrightarrow$: Obtain a new digraph D' by adding a new node z and arcs zv for all $v \in N$. Set the cost of new arcs to be 0. For each $v \in N(D')$, let y_v be the minimum cost among all possible z,v-dipaths. This minimum exists since there is at least one such dipath, and the number of such dipaths is finite.

First, we show if D does not contain a negative dicycle, then y is a feasible potential for D'. Suppose not, i.e., there exist $pq \in A'$ where $\bar{w}_{pq} = w_{pq} + y_p - y_q < 0$. Consider a z,p-dipath P that has minimum cost. By assumption, $w(P) = y_p$. If q is not on P, then $P + pq$ is a z,q-dipath with cost $y_p + w_{pq}$, which is strictly less than y_q by assumption. This contradicts the fact that y_q is the minimum cost of all z,q-dipaths.

Now assume q is on P, and let P_1 be the part of P from z to q, and P_2 be the part of P from q to p. Since P_1 is a z,q-dipath and a minimum cost z,q-dipath has cost y_q, we have $w(P_1) \geq y_q$. Also, $w(P) = w(P_1) + w(P_2) = y_p$. Then $w(P_2) = y_p - w(P_1) \leq y_p - y_q$. By assumption, $y_p - y_q < -w_{pq}$, so $w(P_2) < -w_{pq}$, or $w(P_2) + w_{pq} < 0$. So $P_2 + pq$ is a dicycle whose cost is negative, contradicting the assumption that there is no negative dicycle. Hence y is a feasible potential for D'.

If D does not contain a negative dicycle, then there exists a feasible potential for D'. The same potential applied to only nodes in D is also feasible, hence the dual LP is feasible. Therefore, the original TP is bounded by the objective value of the dual feasible solution. $\square$

1.3.3 Describe the network simplex algorithm for TP.

Given a connected digraph $D = (N, A)$ with arc costs w and node demands b,

1. Find a spanning tree T with a feasible tree flow x by inspection.

2. Calculate the corresponding dual potentials for all $v \in N$.
   a. Pick an arbitrary node, say a, and set $y_a = 0$.
   b. Solve $\bar{w}_{uv} = w_{uv} + y_u - y_v$ for each $uv \in T$ to get potentials for other nodes.
3. Calculate reduced costs $\bar{w}_{uv} = w_{uv} + y_u - y_v$ for $uv \notin T$.
4. If every non-basic arc has non-negative reduced cost, we are done. Otherwise, let uv be a non-basic arc with negative reduce cost.
   a. Form and orient a cycle C by adding uv to T.
   b. If all arcs in C are forward, stop. This is a negative dicycle and the problem is unbounded. Otherwise, let pq be a backward arc with minimum flow in C.
   c. Push x_{pq} units along C and update $T \leftarrow T - uv + pq$.
   d. Go back to (2).

1.3.4 Describe the auxiliary TP and initialization procedure for network simplex.

Given a digraph $D = (N, A)$ and demands $b \in \mathbb{R}^N$, the auxiliary digraph $D' = (N', A')$ with $b' \in \mathbb{R}^{N'}$ and arc cost $w' \in \mathbb{R}^{A'}$ is defined as

- $N' = N \cup \{z\}$.
- $A' = A \cup \{vz \mid v \in N, b(v) < 0\} \cup \{zv \mid v \in N, b(v) > 0\}$.
- $b'(z) = 0$, $b'(v) = b(v)$ for all $v \in N$.
- $w'(e) = 0 \iff e \in A$ and $w'(e) = 1 \iff e \in A' \setminus A$.

Intuitively, we added a new node z with demand 0 and added an arc from each source node to z and from z to each demand node; original nodes have the same demands; original arcs have no cost and new arcs have 1 cost.

The auxiliary TP has a natural feasible flow, i.e., transport all supplies to z then distribute to demand nodes from z, which gives us a starting point for Simplex.

1.4 Characterization of Infeasibility

1.4.1 Prove: A TP with digraph $D = (N, A)$ and node demand $b \in \mathbb{R}^N$ is infeasible if and only if there exists $S \subseteq N$ such that $b(S) < 0$ and $\delta(S) = \varnothing$. [Or equivalently, $b(S) > 0$ and $\delta(\bar{S}) = \varnothing$].

Intuitively, if there is a set of nodes with negative net demand but no leaving arcs, or with positive net demand but no incoming arcs, then the TP is feasible.

$\Longrightarrow$: Suppose the TP is infeasible, so the auxiliary TP has an optimal solution with strictly positive optimal value.

Let x^* be an optimal solution. Let $y_z = 0$. Partition N into $S\_$ and S_+ based on the dual potential. We claim that $S\_ := \{v \in N : y_v = -1\}$ satisfies the conditions $b(S\_) < 0$ and $\delta(S\_) = \varnothing$.

Suppose $uv \in \delta(S\_)$ so that $u \in S\_$ and $v \in S_+$. Then $\bar{w}_{uv} = w_{uv} + y_u - y_v = 0 - 1 - 1 = -2 < 0$, contradicting the maximality of x^*. Thus, $\delta(S\_) = \varnothing$.

For any $uv \in \delta(\bar{S}\_)$, $\bar{w}_{uv} = w_{uv} + y_u - y_v = 0 + 1 + 1 = 2 \neq 0$ so uv is non-basic and hence $x_{uv} = 0$. Since the optimal value is positive, some flow goes from $S\_$ to z, so

$$0 > x(\delta(\bar{S}\_)) - x(\delta(S\_)) = \sum_{v \in S\_} (x(\delta(\bar{v})) - x(\delta(v))) = \sum_{v \in S\_} b_v = b(S\_).$$

$\Longleftarrow$: Let $S \subseteq N$ where $b(S) < 0$ and $\delta(S) = \varnothing$. Suppose for a contradiction that x is a feasible flow. By feasibility,

$$x(\delta(\bar{S})) - x(\delta(S)) = \sum_{v \in S} (x(\delta(\bar{v})) - x(\delta(v))) = \sum_{v \in S} b_v = b(S) < 0.$$

By assumption, $\delta(S) = \varnothing \implies x(\delta(S)) = 0$ so $x(\delta(\bar{S})) < 0$, but $x \geq \mathbf{0} \implies x(\delta(\bar{S})) \geq 0$, a contradiction. $\square$

2 Minimum Cost Flow Problem

2.1 Minimum Cost Flow Problem

2.1.1 Describe the minimum cost flow problem.

Given digraph $D = (N, A)$, node demands $b \in \mathbb{R}^N$, arc costs $w \in \mathbb{R}^A$, arc capacities $c \in \mathbb{R}^A_+$, we want to find a feasible flow x satisfying node demands and capacity constraints while minimizing the total cost.

2.1.2 What are the primal and dual LP for TP (explicit equations, no matrix)?

The primal and dual LPs are given by

$$\begin{array}{ll} \min & w^T x \\ s.t. & x(\delta(\bar{v})) - x(\delta(v)) = b_v \quad \forall v \in N \\ & x_e \leq c_e \quad (\forall e \in A) \\ & x \geq \mathbf{0} \end{array}$$

$$\begin{array}{lll} \max & b^T y - c^T z & \\ s.t. & y_v - y_u - z_{uv} \leq w_{uv} & \forall uv \in A \\ & z_{uv} \geq 0 & \forall uv \in A \end{array}$$

2.1.3 Describe the constraint matrix for MCFP.

The constraint matrix is given by

$$\begin{array}{c|cc} & |A| & |A| \\ \hline |N| & M & O \\ |A| & I & I \end{array} \quad \begin{pmatrix} M & O \\ I & I \end{pmatrix} x = \begin{pmatrix} b \\ c \end{pmatrix},$$

where M is the incidence matrix.

2.1.4 Prove: A tree flow in MCFP consists of a spanning tree T and a feasible flow x where all arcs not in T satisfies $x_e \in \{0, c_e\}$.

The constraint matrix has rank $|N| - 1 + |A|$.

Recall from TP that the incidence matrix has rank $|N| - 1$. Observe the $|A|$ capacity constraints are linearly dependent, so we can extend the basis by adding them in. The rank of constraint matrix is thus $|N| - 1 + |A|$.

For each arc $e \in A$, at least one of x_e and s_e is in the basis.

Recall that non-basic variables are set to zero. Suppose for some $e \in A$, both x_e and x_e are both non-basic, so $x_e = s_e = 0$. Then there is no way to satisfy the capacity constraint for e.

Therefore, for each arc $e \in A$, there are three possibilities:

1. Only x_e is in the basis. (The arc has flow equal to capacity.)
2. Only s_e is in the basis. (The arc does not have flow at all.)
3. Both x_e and s_e are in the basis. (The arc has positive flow, but less than its capacity.)

There exist $|N| - 1$ arcs where both x_e and s_e are in the basis.

Let k denote the number of type-3 arcs. There are $|A|$ arcs in total, so there are $|A| - k$ type-1/2 arcs. We do a double-counting.

Recall the rank of constraint matrix is $|N| - 1 + |A|$, so a basis has $|N| - 1 + |A|$ variables. Each of the k type-3 arcs contributes 2 basic variables while each of the type-1/2 arcs contribute 1 basic variable. Then $2k + |A| - k = |N| - 1 + |A| \implies k = |N| - 1$.

Arcs where both x_e and s_e are in the basis cannot form a cycle.

Suppose a subset of type-3 arcs form a cycle. Consider the constraint matrix corresponding to the cycle. We could multiply -1 to appropriate columns of the left half to turn M into an incidence matrix of a dicycle, so the sum of entries of the first $|N|$ rows are zero, then multiply -1 to appropriate columns of the right half so the sum of entries of the rest $|A|$ rows are also zero. It follows that the columns of the matrix are linearly dependent. This is a contradiction as a subset of a basis cannot be linearly dependent.

Conclusion.

Since there are $|N|$ nodes in total and there are $|N| - 1$ type-3 arcs which contains no cycle, it follows that these type-3 arcs correspond to a spanning tree.

For arcs in the spanning tree T, both x_e and s_e are in the basis, so $0 < x_e, s_e < c_e$. For non-basic arcs e, if x_e is in the basis, then $s_e = 0 \implies x_e = c_e$; if s_e is in the basis, then $x_e = 0$ and $s_e = c_e$.

2.1.5 What's the complementary slackness condition for MCFP?

From primal and dual LP, the CS conditions are

1. $z_{uv} > 0 \implies x_{uv} = c_{uv}$, and
2. $-y_u + y_v - z_{uv} < w_{uv} \implies x_{uv} = 0$.

Rewrite the dual constraints as $-z_{uv} \leq \bar{w}_{uv}$ or $z_{uv} \geq -\bar{w}_{uv}$ and $z \geq \mathbf{0}$. The objective function contains $-c^T z$ where $c > 0$, so maximizing $b^T y - c^T z$ is equivalent to minimizing z. Therefore, $z_{uv} = \max\{0, -\bar{w}_{uv}\}$.

For (1): If $z_{uv} > 0$, then we must have $-\bar{w}_{uv} > 0$ or $\bar{w}_{uv} < 0$. Thus, we can rewrite (1) as $\bar{w}_{uv} < 0 \implies x_{uv} = c_{uv}$.

For (2): If $-y_u + y_v - z_{uv} < w_{uv}$, then $z_{uv} > -\bar{w}_{uv}$, so $\bar{w}_{uv} > 0$ and $z_{uv} = 0$. Thus, we can rewrite (2) as $\bar{w}_{uv} > 0 \implies x_{uv} = 0$.

Hence, the optimality conditions for MCFP are:

1. $\bar{w}_{uv} < 0 \implies x_{uv} = c_{uv}$.
2. $\bar{w}_{uv} > 0 \implies x_{uv} = 0$.
3. $0 < x_{uv} < c_{uv} \implies \bar{w}_{uv} = 0$.

2.1.6 Describe the network simplex for MCFP.

Given a connected digraph $D = (N, A)$ with arc costs w and node demands b,

1. Find a spanning tree T with a feasible tree flow x by inspection.
2. Calculate the corresponding dual potentials for all $v \in N$.
    a. Pick an arbitrary node, say a, and set $y_a = 0$.
    b. Solve $\bar{w}_{uv} = w_{uv} + y_u - y_v$ for each $uv \in T$ to get potentials for other nodes.
3. Calculate reduced costs $\bar{w}_{uv} = w_{uv} + y_u - y_v$ for $uv \notin T$.
4. Find a non-basic arc uv where either (a) $\bar{w}_{uv} < 0$ and $x_{uv} = 0$, or (b) $\bar{w}_{uv} > 0$ and $x_{uv} = c_{uv}$. If no such arc exists, the current solution is optimal.
    a. Form and orient a cycle C by adding uv to T.
    b. Orient C in the direction of uv if (a); orient C in the direction opposite of uv if (b).
    c. Let $t := \min(\{c_f - x_f : f \text{ is a forward arc of } C\} \cup \{x_f : f \text{ is a backward arc of } C\})$.
    d. Push t units of flow along C. Update T and go back to (2).

2.1.7 Prove: An MCFP is infeasible if and only if there exists $S \subseteq N$ such that $b(S) > c(\delta(\bar{S}))$ or $b(S) < -c(\delta(S))$.

Intuitively, if there exists $S \subseteq N$ where total demand is more than total capacity of in-arcs or total supply is more than total capacity of out-arcs, then the MCFP is infeasible.

$\implies$: We use the same auxiliary digraph as TP. Suppose our MCFP is infeasible, then the auxiliary MCFP has optimal value strictly positive. Let x be a feasible tree flow. Set potentials y with $y_z = 0$ and all other potentials are either 1 or -1. Define S_+ and $S\_$ as before. We will show $S\_$ satisfies $b(S\_) < -c(\delta(S\_))$.

Consider e from $S\_$ to S_+ and f from S_+ to $S\_$. By CS conditions,

- $\bar{w}_e = 0 - 1 - 1 = -2 < 0 \implies x_e = c_e$, so all arcs going from $S\_$ to S_+ are at full capacity.
- $\bar{w}_f = 0 + 1 + 1 = 2 > 0 \implies x_f$, so there is no in-flow from S_+ to $S\_$. Since there is also no arc from z to $S\_$, $S\_$ has no in-flow at all.

Since the auxiliary MCFP has strictly positive optimal value and $b_z = 0$, there must also be some flow leaving $S\_$ for z. Thus,

$$b(S\_) = x(\delta(\bar{S}\_)) - x(\delta(S\_)) = 0 + \sum_{u \in S\_, v \in S_+} (-c_{uv}) + \sum_{w \in S\_} (-x_{wz}) < -c(\delta(S\_)).$$

$\Longleftarrow:$ Suppose there exists $S \subseteq N$ st $b(S) > c(\delta(\bar{S}))$. Suppose for a contradiction that there is a feasible flow x, then

$$b(S) = \sum_{v \in S} b_v = \sum_{v \in S} [x(\delta(\bar{v})) - x(\delta(v))] = x(\delta(\bar{S})) - x(\delta(S)) \leq c(\delta(\bar{S})) - 0 = c(\delta(\bar{S})),$$

which contradicts the hypothesis $b(S) > c(\delta(\bar{S}))$. $\square$

2.2 MCFP Applications

2.2.1 Describe how you could solve the minimum cost perfect matching problem using MCFP.

Given a bipartite graph $G = (V, E)$, $V = A \cup B$, $|A| = |B|$, and edge costs $w \in \mathbb{R}^E$, we want to find a perfect matching in G of minimum total cost. We will formulate this as a MCFP:

- Direct each edge from A to B.
- Set the capacity for each arc to be 1.
- Set $v \in A$ as supply nodes with $b_v = -1$ and $v \in B$ as demand nodes with $b_v = 1$.
- Arc costs stay the same.

2.2.2 Prove: If a MCFP has an optimal solution, and all capacities and node demands are integers, then there exists an integral optimal solution.

Suppose first that we have an integral flow. At each step of network Simplex, we chose $t := \min(\{c_f - x_f : f \text{ is a forward arc of } C\} \cup \{x_f : f \text{ is a backward arc of } C\})$ where C is the unique cycle created with the entering arc. Since c and x are integral, t is integral, so the new flow created by pushing t along C is integral.

Now, because b is integral, we have an integral feasible flow (by sending all flow from supply nodes to z then distribute flow to demand nodes) for the auxiliary digraph. If we run Simplex with this flow, the flow remains integral.

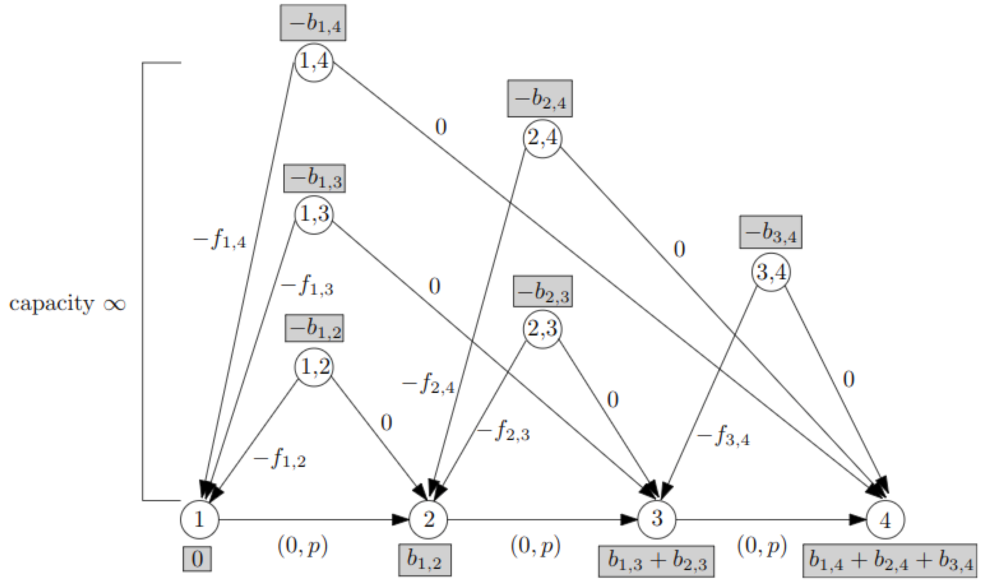
2.2.3 Justify your solution, that is, a solution of the MCFP instance is an optimal matching.

For our MCFP formulation, by the lemma above, there exists an integral optimal solution x. Then for each arc $e \in A$, $x_e \in \{0, 1\}$. The set of active arcs $M = \{e : x_e = 1\}$ is a perfect matching because each node is incident with exactly one edge (supply is 1 so only one arc can be chosen). Also, any perfect matching corresponds to an integral flow. Hence, our MCFP solves the MCPM problem.

2.2.4 Describe how you could solve the airline scheduling problem using MCFP. (More importantly, you should be able to draw a diagram with $n = 4$ cities.)

Consider the following airline scheduling problem:

- A plane visits cities $1, 2, \ldots, n$ in this order.
- There are $b_{i,j}$ passengers from city i to city j $(i < j)$.
- The ticket costs are $f_{i,j}$ $(i < j)$.
- The plane has capacity P.
- Our goal is to maximize ticket costs subject to plane capacity.



The arcs between nodes represent the path of the plane with capacity p, cost 0.

- Cost 0: We need to make the trip anyway, so we consider the cost as 0.
- Capacity p: the plane has capacity p, so each trip has p people at most.

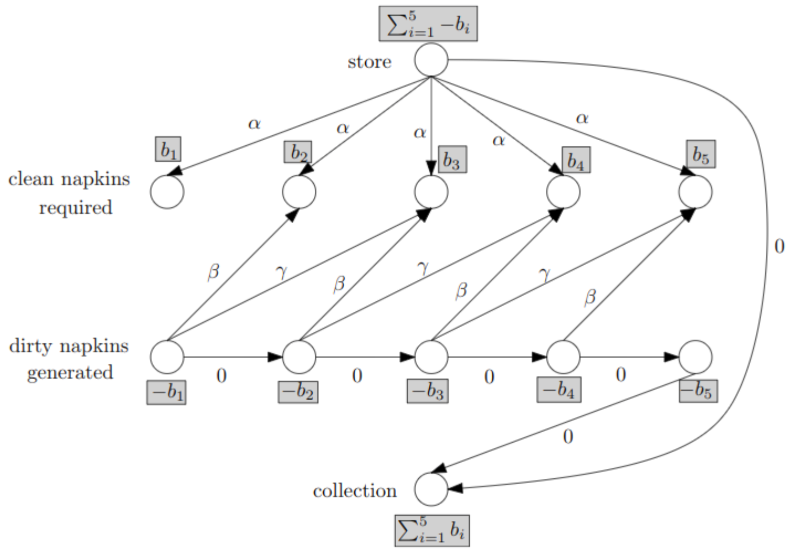
Each node (i, j) takes passengers from i to j, either through the plane, or through other means.

- $-f_{i,j}$ cost from node (i, j) to node (i): make money (negative since minimization).
- 0 cost from node (i, j) to node (j): passengers not taking the plane, no gain.

2.2.5 Describe how you could solve the catering problem using MCFP. (More importantly, you should be able to draw a diagram with $n = 5$ days.)

Consider the following catering problem:

- A caterer requires b_i clean napkins for each day $i = 1, \dots, n$.
- They can buy new ones from the store for a cost of α.
- Used napkins can be washed in two ways:
  - 1-day service for a cost of β each.
  - 2-day service for a cost of γ each.
  - Used napkins can be kept in storage for free.
- We want to minimize the total cost of napkins.



2.2.6 Describe how you could transform a matrix with consecutive ones into an incidence matrix.

Suppose $A \in \mathbb{Z}^{m \times n}$ where each column has consecutive 1's and all other entries are 0's. We could do the following transformation:

1. Add slack variables (appending $-I_m$ to the right of A.)
2. Add a redundant row of all zeros.
3. Subtract the $(m-1)$-th row from m-th, then subtract the $(m-2)$-th from $(m-1)$th, etc.

Observe A' becomes an incidence matrix! The entry with "1" in A' corresponds to the topmost "1" in A and "-1" in A' corresponds to the entry one below the bottommost "1" in A. Also, the demands add up to zero.

3 Shortest Dipath Problem

3.1 Shortest Dipath Problem

3.1.1 Describe the shortest dipath problem.

Given a dipath $D = (N, A)$, arc costs $w \in \mathbb{R}^A$, two distinct nodes $s, t \in N$, we wish to find a minimum cost s, t-dipath.

3.1.2 What are the primal and dual LP for shortest dipath problem?

The primal LP is given by

$$\begin{aligned} \min \quad & w^T x \\ s.t. \quad & x(\delta(\bar{v})) - x(\delta(v)) = \begin{cases} -1 & v = s \\ 1 & v = t \\ 0 & \text{else} \end{cases} \\ & x \geq \mathbf{0} \end{aligned}$$

The dual LP is identical to the one for TP:

$$\begin{aligned} \max \quad & y_t - y_s \\ s.t. \quad & y_v - y_u \leq w_{uv} \quad \forall uv \in A \end{aligned}$$

3.1.3 Define: a characteristic vector of a path.

A characteristic vector of a path P is a vector $x^P \in \mathbb{R}^A$ where $x^P_e = 1$ if $e \in P$ and 0 otherwise.

3.1.4 Prove: If $\bar{x}$ is an integral feasible solution to LP, then $\bar{x}$ is a sum of the characteristic vector of an s, t-dipath and a collection of dicycles.

Consider the set of active arcs $F = \{e \in A : \bar{x}_e > 0\}$ in $\bar{x}$.

We first show the existence of an s, t-dipath. Let $\delta(S)$ be an s, t-cut. The net flow of S is -1 (because the net flow of $s \in S$ is -1 and the net flow of all other nodes in $\delta(S)$ are zero by construction), so there must be at least one arc in $\delta(S)$ with non-zero flow, i.e., it is in F. Since this holds for every s, t-cut, there exists an s, t-dipath P using arcs in F.

We now show the possible existence of a collection of dicycles. Consider the flow obtained by removing the characteristic vector of P from the integral feasible solution $x' := \bar{x} - x^P$. Since both $\bar{x}, x^P$ satisfy the flow constraints, we get $x'(\delta(\bar{v})) - x'(\delta(v)) = 0$ for all $v \in N$.

Consider the set of active arcs $F' := \{e \in A : x'_e > 0\}$ in x'. If $F' = \varnothing$, we are done as $\bar{x}$ was an s, t-dipath. Otherwise, take a longest dipath $v_1, \ldots, v_k$ in F'. Since $x'(\delta(\bar{v})) - x'(\delta(v)) = 0$, there is an arc $v_k v_i$ for some $i < k$. Moreover, v_i cannot be outside of the path since we took a longest dipath. This forms a dicycle $C := v_i, v_{i+1}, \ldots, v_k, v_i$.

Removing this cycle from flow x', we get $x'' := x' - x^C$, which also satisfies $x''(\delta(\bar{v})) - x''(\delta(v)) = 0$ for all $v \in N$ and the sum of the flow has decreased by at least 1. By induction, we are done. $\square$

3.1.5 Prove: If there are no negative dicycles, then our LP formulation has an optimal solution that is the characteristic vector of an s,t-dipath.

Let x be an optimal solution. By the previous proposition, $x = x^P + x^{C_1} + \cdots + x^{C_k}$ for some s,t -dipath P and dicycles $C_1, \ldots, C_k$; the cost of x is $w^T x = w^T x^P + w^T x^{C_1} + \cdots + w^T x^{C_k}$.

We now show all cycles have zero cost, i.e., $w^T x^{C_i} = 0$. Suppose not. We are given that there are no negative dicycles, so suppose C_i has positive cost. Then $x' := x - x^{C_i}$ is a feasible solution with cost $w^T x' = w^T x - w^T x^{C_i} < w^T x$, contrary to the optimality of x.

Thus, all dicycles have zero cost and $w^T x = w^T x^P$. It follows that x^P is an optimal solution that is a characteristic vector of an s,t-dipath.

3.2 Ford's Algorithm

3.2.1 Prove: Let $D = (N, A)$ be a digraph with arc cost $w \in \mathbb{R}^A$ with no negative dicycle. If $v_1, \ldots, v_k$ is a shortest v_1, v_k-dipath, then $v_1, \ldots, v_i$ is a shortest v_1, v_i-dipath.

Since there are no negative dicycles, the LP has an optimal integral solution corresponding to a characteristic vector of an v_1, v_k-dipath. Then there is an optimal dual solution y where all arcs of P are equality arcs by CS conditions. Let $P' := v_1, \ldots, v_i$. Note that y is still feasible for the dual LP of the shortest v_1, v_i-dipath problem. Moreover, any arc in P' is also in P, so all the arcs of P' are equality arcs with respect to y. Thus, the CS conditions are satisfied for the v_1, v_i-dipath problem and P' is an optimal solution. $\square$

3.2.2 Define: A rooted tree at s.

A tree T is rooted at s if for all $t \in N(T)$, the unique s, t-path in T is an s, t-dipath. Note that:

1. There is an s, t-dipath in D for all $t \in N$ iff there is a spanning tree in D rooted at s.
2. Let T be a spanning tree in D. Then T is rooted at s iff $|\delta(\bar{s})| = 0$ and $|\delta(\bar{v})| = 1$ for $v \in N(T) \setminus \{s\}$.

3.2.3 What does Ford's algorithm accomplish?

Ford's algorithm allows us to find shortest s, v-dipath for all $v \in N$ in one go.

3.2.4 Describe Ford's algorithm.

Assume every node can be reached from s via a dipath, the algorithm tries to produce a feasible potential and a rooted spanning tree T at s, so that the arcs of T are all equality arcs. At each step, we keep track of the potential and predecessor of each node.

0. Initialization.
    a. Set $y_s = 0$ and $y_v = \infty$ for all $v \in N \setminus \{s\}$.
    b. Set predecessor $p_v = \varnothing$ for all $v \in N$.
1. Correction. While y is not feasible, i.e., there exists an arc uv with a negative reduced cost,
    a. Set $y_v = y_u + w_{uv}$, so that uv becomes an equality arc.
    b. Set $p_v = u$. Loop.

A key observation is by setting $y_v = y_u + w_{uv}$ where $y_v > y_u + w_{uv}$ to begin with, we are decreasing y_v.

3.3 Correctness and Termination of Ford's Algorithm

3.3.1 Define: Predecessor graph.

The predecessor graph D_p is given by $N(D_p) := N$ and $A(D_p) : \{p_v v : v \in N\}$.

3.3.2 Prove: Throughout the algorithm, $\bar{w}_e \leq 0$ for all arcs $e \in A(D_p)$.

Let $v \in N$ be arbitrary and p_v be its predecessor. When a correction takes place with an arc whose head is v, $\bar{w}_{p_v v} \leftarrow 0$. Until the predecessor of v is changed again, the reduced cost stays non-positive, and only y_{p_v} can change (due to connecting other arcs). By observation above, y_{p_v} can only decrease, so $\overline{w}_{p_v v} = w_{p_v v} - y_v + y_{p_v}$ only decreases. □

3.3.3 Prove: Let $D = (N, A)$ be a digraph with weights $w \in \mathbb{R}^A$ and feasible potentials $y \in \mathbb{R}^N$. Let Q be an s,t-diwalk. Then $w(Q) \geq y_t - y_s$. Moreover, $w(Q) = y_t - y_s$ iff every arc of Q is an equality arc.

Let $Q = v_1, \ldots, v_k$ where $s = v_1$ and $t = v_k$. Since y is feasible, $y_{v_{i+1}} \leq y_{v_i} + w_{v_i v_{i+1}}$ for $i \in \{1, \ldots, k-1\}$. Adding them up, we get $y_{v_2} + \cdots + y_{v_k} \leq y_{v_1} + \cdots + y_{v_{k-1}} + w_{v_1 v_2} + \cdots + w_{v_{k-1} v_k}$. Subtracting $y_{v_1} + \cdots + y_{v_{k-1}}$ from both sides, we get $y_{v_k} - y_{v_1} = y_t - y_s \leq w(Q)$. Moreover, $y_t - y_s = w(Q)$ iff $y_{v_{i+1}} = y_{v_i} + w_{v_i v_{i+1}}$ hold for $i \in \{1, \ldots, k-1\}$. □

3.3.4 Prove: Let $D = (N, A)$ be a digraph with weight $w \in \mathbb{R}^A$. If D has a negative dicycle then D has no feasible potentials.

Let $y \in \mathbb{R}^N$ be feasible and $C = v_1 v_2, \ldots, v_{k-1} v_k, v_k v_1$ be a dicycle. Now C is a v_1, v_1-diwalk, so we must have $w(C) \geq y_{v_1} - y_{v_1} = 0$. Hence, D has no negative dicycle. □

3.3.5 Prove: If D_p contains a dicycle (at any point in the algorithm), then D contains a negative dicycle and the algorithm does not terminate.

Suppose we produce a dicycle $C := (v = v_1), v_2, \ldots, (v_k = u), v$ in D_p by connecting arc uv. Then it must be true that in the previous iteration, $\bar{w}_{uv} = w_{uv} + y_u - y_v < 0$. By proposition above, $\bar{w}_e \leq 0$ for all $e \in A(D_p)$ throughout the algorithm, so $\overline{w}_{v_i v_{i+1}} = w_{v_i v_{i+1}} + y_{v_i} - y_{v_{i+1}} \leq 0$ for $1 \leq i \leq k$. Since C is a dicycle, adding up these inequalities cancel out y's and we are left with $\sum_{i=1}^{k-1} w_{v_i v_{i+1}} + w_{v_k v_1} < 0$, i.e., C is a negative dicycle. It follows from the previous lemma that there cannot be a feasible potential, so the algorithm never terminates. □

3.3.6 Prove: Suppose the algorithm terminates. Then D_p is a spanning tree of shortest dipaths rooted at s. Furthermore, y_v is the cost of a shortest s,v-dipath.

Since the algorithm terminates, D_p cannot contain a cycle and s does not have a predecessor. Then D_p is a rooted spanning tree. Since all nodes other than s has a predecessor, $|\delta(\bar{v})| = 1$ for all $v \in N \setminus \{s\}$ and D_p is rooted at s. Now, all arcs in D_p are equality arcs, because $\overline{w}_e \leq 0$ for all $e \in A(D_p)$ and $\overline{w} < 0$ is impossible since y is feasible (by termination).

For $v \in N$, let P be the unique s, v-dipath in D_p. Consider the LP formulation of the shortest s, v -dipath problem: x^P is feasible for the primal and y is feasible for the dual. Since all arcs in P are equality arcs, CS conditions hold, so x^P is optimal and the objective of the dual is $y_v - y_s = y_v - 0 = y_v$, i.e., y_v is the cost of a shortest s, v-dipath. $\square$

3.4 Bellman-Ford's Algorithm

3.4.1 Describe Bellman-Ford's algorithm.

The idea is to go through arcs in "passes".

0. Initialization.
    a. Set $y_s = 0$ and $y_s = \infty$ for all $v \in N \setminus \{s\}$.
    b. Set predecessor $p_v = \varnothing$ for all $v \in N$.
    c. Set the counter $i = 0$.
1. Correction. While $i < |N| - 1$,
    a. For each $uv \in A$, if $\bar{w}_{uv} < 0$, set $y_v = y_u + w_{uv}$ and $p_v = u$.
    b. Increment i.

3.4.2 Prove: Let d_v denote the *cost* of a shortest s, v-dipath. Suppose D does not have any negative dicycle. Then at any point in the algorithm $y_v \geq d_v$.

The claim is clearly true at initialization. If $y_v \neq \infty$, then there exists a dipath from s to v using D_p. For each of these arcs e, $\bar{w}_e \leq 0$. Adding up all inequalities $\bar{w}_e \leq 0$ for all arcs e in this s, v-dipath, we obtain $w(P) - y_v \leq 0$. Since d_v is the cost of a shortest s, v-dipath, $d_v \leq w(P) \leq y_v$ are we are done. $\square$

3.4.3 Prove: Suppose no negative dicycles exist. After the ith iteration, if there is a shortest s, v-dipath using at most i arcs, then $y_v = d_v$.

We do an induction on i. When $i = 0$ (initialization), trivial. Assume that this is true after the i th iteration. We want to show this still holds after $(i + 1)$th iteration.

Pick v which has a shortest s, v-dipath that uses at most $(i + 1)$ arcs. If there is a shortest s, v-dipath that uses at most i arcs, by induction hypotheses, $y_v = d_v$. By proposition, y_v will not change.

Suppose there is a shortest s, v-dipath that uses $(i + 1)$ arcs, say $s = v_1, \ldots, v_{i+1} = v$. Since no negative dicycle exists, $v_1, \ldots, v_i$ is a shortest s, v_i-dipath that uses i arcs. By induction, $y_{v_i} = d_{v_i}$ after the ith iteration and this does not change after the $(i + 1)$-th iteration.

Consider $\bar{w}_{v_i v_{i+1}}$.

- If $\bar{w}_{v_i v_{i+1}} = 0$, this means that $y_{v_{i+1}} = y_{v_i} + w_{v_i v_{i+1}} = d_{v_i} + w_{v_i v_{i+1}} = w(P) = d_{v_{i+1}}$.
- If $\bar{w}_{v_i v_{i+1}} > 0$, this means that $y_{v_{i+1}} < y_{v_i} + w_{v_i v_{i+1}} = d_{v_i} + w_{v_i v_{i+1}} = w(P) = d_{v_{i+1}}$. Contradiction. This cannot happen.
- If $\bar{w}_{v_i v_{i+1}} < 0$, this means that the algorithm will correct the arc $v_i v_{i+1}$ in the $(i + 1)$th iteration so $y_{v_{i+1}} = y_{v_i} + w_{v_i v_{i+1}} = d_{v_{i+1}}$.

We are done by induction. $\square$

3.4.4 Prove: At the end of Bellman-Ford, if y is feasible, then $y_v = d_v$ for all $v \in N$. Otherwise, you can conclude there exists a negative dicycle.

Bellman-Ford runs $|N| - 1$ iterations. Any shortest s, v-dipath could use at most $|N| - 1$ arcs. If y is feasible, then there are no negative dicycles. By the theorem above, $y_v = d_v$ for all $v \in N$. If not, then there exists a negative dicycle.

3.5 Dijkstra's Algorithm

3.5.1 Describe the intuition of Dijkstra's algorithm.

If there are no negative arcs, $y = 0$ is a feasible potential for the dual. We wish to raise potentials by t for non-tree nodes while maintaining feasibility.

Let T be our current tree.

1. $u, v \notin N(T)$: both y_u and y_v increases by t, so $\bar{w}_{uv} = w_{uv} - y_v + y_u$ stays the same.
2. $u, v \in N(T)$: we do not change the potentials, so $\bar{w}_{uv}$ stays the same.
3. $u \notin N(T) \wedge v \in N(T)$: $\bar{w}_{uv}$ increases and that does not affect feasibility of the potentials.
4. $u \in N(T) \wedge v \notin N(T)$: $\bar{w}_{uv}$ decreases by t; thus choose t to be minimum among all such arcs.

Now, the arc which determined the minimum becomes an equality arc and we can add it to T.

3.5.2 Describe Dijkstra's algorithm.

1. Initialize $y_v = 0$ for all v and $T = (\{s\}, \varnothing)$.
2. While T is not a spanning tree,
    a. Pick $u \in \delta(N(T))$ such that $\bar{w}_{uv} = \min\{\bar{w}_e : e \in \delta(N(T))\}$.
    b. Update $y_z := y_z + \bar{w}_{uv}$ for all $z \notin N(T)$.
    c. Add uv to $A(T)$ and u to $N(T)$.

3.5.3 Prove its correctness.

By our work above, y is always feasible. This includes at initialization as we do not have negative costs. All arcs in T are equality arcs. In addition, the algorithm produces a spanning tree rooted at S. Thus, the same LP argument gives that it must be a tree of shortest s, v-dipaths for all $v \in N$.

3.6 Application of Shortest Dipath

3.6.1 Describe how we could solve network reliability using shortest dipath.

A network $D = (N, A)$ where each arc e is assigned an associated reliability $r_e \in (0, 1]$. Think of this as a probability that r_e is operational. For a given dipath P, the reliability of P is $r(p) = \prod_{e \in P} r_e$. Our goal is to maximize reliability amongst all s, t-dipaths.

Notice that $\log r(P) = \sum_{e \in P} \log r_e$ and $\log$ is strictly increasing so it suffices to compare logarithms of reliability. We also make this a minimization problem by having negative arc costs: let $-\log r_{ij}$ denote the cost of arc ij. We could then solve it using Bellman-Ford.

3.6.2 Describe how we could solve currency exchange problem using shortest dipath.

We have a set of currencies. There is an exchange rate r_{uv} representing how much does 1 unit of currency u converts into currency v. We want to exchange a series of currencies back to the original one so that we make a profit.

Since we can make some profit, the following inequality must hold:

$$\prod_{e \in C} r_e = r_{v_1, v_2} r_{v_2, v_3} \ldots r_{v_k, v_1} > 1 \iff \log \prod_{e \in C} r_e > 0 \iff \sum_{e \in C} (-\log r_e) < 0$$

Label each arc with cost $-\log r_e$. We can just run Bellman-Ford.