

Kliment Mamykin, UNI 2770

Algorithms for Data Science, Homework 3

Problem 1

See last page

Problem 2

i. Derive a necessary condition for a feasible circulation with demands to exist.

Given the demand constraints $v \in V$, $f^{in}(v) - f^{out}(v) = d_v$ we can sum up both sides of the equation for all nodes in the network:

$$\sum_{v \in V} (f^{in}(v) - f^{out}(v)) = \sum_{v \in V} d_v$$

Since each directed edge contributes the same in-flow into some node and the same out-flow from another node, we can expand and re-arrange all terms on the left side, combining together f^{in} and f^{out} of the same edge.

$$\sum_{(u,v) \in E} (f^{in}(v) - f^{out}(u)) = 0 = \sum_{v \in V} d_v$$

So the necessary condition for a circulation to exist is $\sum_{v \in V} d_v = 0$

ii. The problem can be transformed into Max Flow problem by augmenting the network in the following way:

- Add a source node s and a sink node t
- Add edges from s to each node in S with capacity $c = |d_v|$. Since d_v for source nodes is negative, meaning in-flow is less than out-flow, adding an extra in-flow from s "plugs the hole", preserving the flow.
- Add edges from each node in T to t node, with capacity $c = |d_v|$. This similarly preserves the flow at each node.

With these augmentations we convert the demand network into a flow graph, converting the demand constraint into the flow preservation constraint.

A feasible demand circulation will exist iff there exists an s-t flow in the augmented graph with $\max |f| = \sum_{v \in S} |d_v|$

Problem 3

(a) Show that a min-cost problem can be converted to a max flow problem.

Here we also need to augment the supply network by introducing two nodes s and t , and connecting s with each source node v with an edge with capacity $c_{s,v} = s_v$. Then connect each sink node v with t with an edge with capacity $c_{v,t} = |s_v|$. All introduced edges would have 0 cost to exclude them from affecting the min-cost solution. This converts the supply network to s-t graph. If there is a flow possible in the supply network, then there must be a max flow in the graph such that all edges coming out of s are filled to capacity.

$$f_{sj} = c_{sj} \quad \forall (s,j) \in E$$

The problem of max-flow in this graph is related to the min-cost problem in the following way. When calculating max flow, on each iteration of the algorithm we compute the residual graph, then pick an augmenting path, and augment the flow. Since augmenting path has a positive non-zero capacity on each iteration the graph flow will increase. When solving a min-cost problem we could compute residual graph for the costs, and pick an augmenting path that reduces the cost of flow, and augment the flow. Then at some point we will reach a min-cost flow.

When calculating residual graph, the reverse flow edges (that undo/push back flow) would have the negative cost of the forward edge (so that edge not only pushes back the flow, but also reduces the cost). I am stuck figuring out how to calculate the augmenting path in this case.

(b) min-cost LP formulation:

Variables: $f_{ij}, (i,j) \in E, m$ variables.

$$\min_{f_{ij} \geq 0, a_{ij} \geq 0} \sum_{(i,j) \in E} a_{ij} f_{ij}$$

Capacity constraints (m constraints):

$$f_{ij} \leq c_{ij}, \forall (i,j) \in E$$

Supply constraint (n constraints):

$$\sum_{j|(v,j) \in E} f_{vj} - \sum_{j|(j,v) \in E} f_{jv} = s_v, \forall v \in V$$

Problem 4

Minimizing the number of rolls cut with known cut patterns

There are only 12 cut patterns a 300cm roll can be cut into rolls of widths 135, 108, 93, 42 minimizing waste/maximizing usage:

Cut Pattern	W1	W2	W3	W4	R1	R2	R3	R4	Usage	Waste
1	135	108	93	42	0	0	0	7	294	6
2	135	108	93	42	0	0	1	4	261	39
3	135	108	93	42	0	0	2	2	270	30
4	135	108	93	42	0	0	3	0	279	21
5	135	108	93	42	0	1	0	4	276	24
6	135	108	93	42	0	1	1	2	285	15
7	135	108	93	42	0	1	2	0	294	6
8	135	108	93	42	0	2	0	2	300	0
9	135	108	93	42	1	0	0	3	261	39
10	135	108	93	42	1	0	1	1	270	30
11	135	108	93	42	1	1	0	1	285	15
12	135	108	93	42	2	0	0	0	270	30

Let $w = [135, 108, 93, 42]$ be an array of widths of desired rolls, $d = [97, 610, 395, 211]$ be an array of the minimum number of desired rolls of corresponding width, R_{ij} be a matrix of number of resulting rolls of each width possible to cut with minimum waste, where i is a cut pattern index and j is the index of the roll width used.

Then the problem can be formalized as an LP problem:

Variables: x_i , $1 \leq i \leq 12$ (12 variables) the number of 300cm rolls cut with cut pattern i .

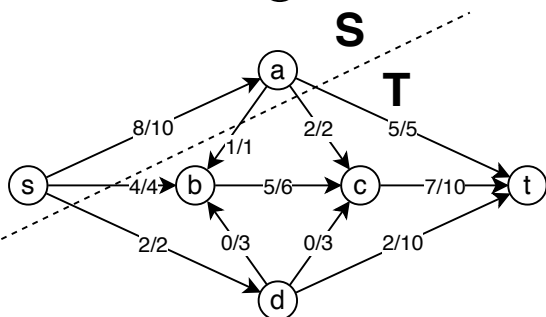
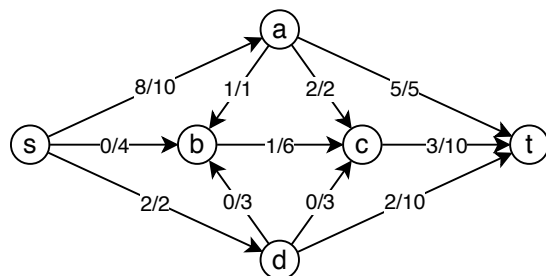
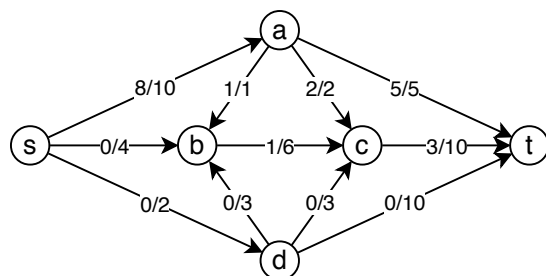
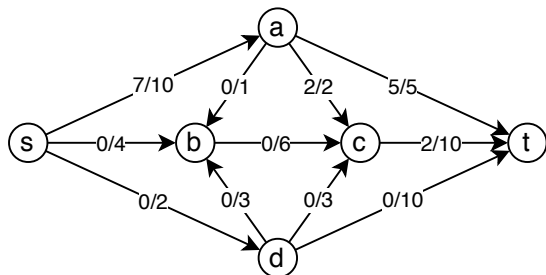
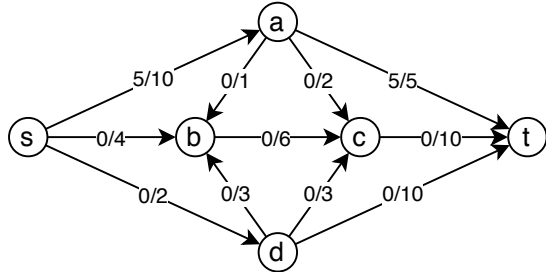
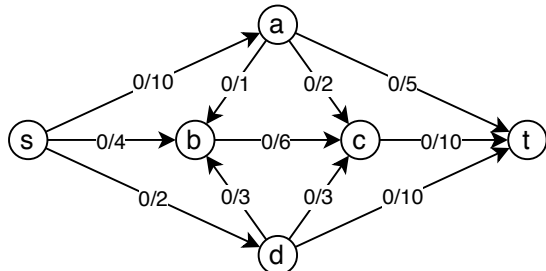
Objective function - minimize the number of rolls used:

$$\min_{x_i \geq 0} \sum_{i=1}^{12} x_i$$

Constraints - satisfy customer demand:

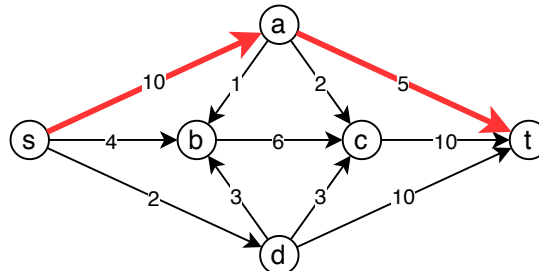
$$\sum_{i=1}^{12} x_i R_{ij} \geq d_j \quad 1 \leq j \leq 4$$

Graph and flow at each iteration

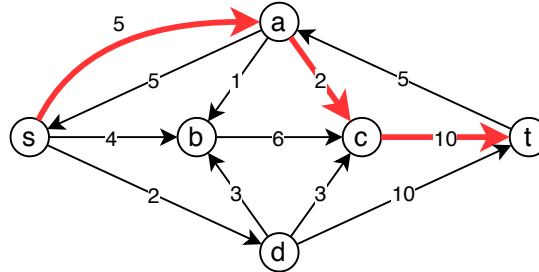


max flow = 14

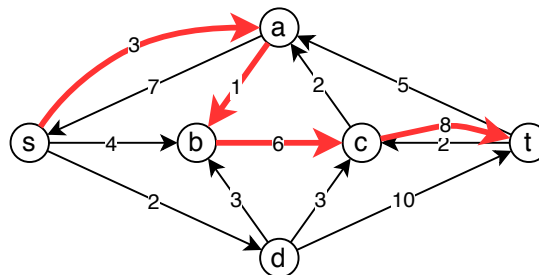
Residual Graph & Augmenting Path



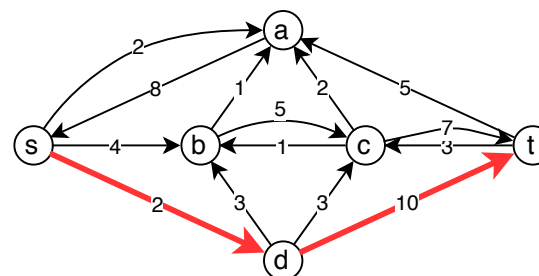
Augmenting paths:
 $c(\{s,a,t\}) = 5$ <- pick this
 $c(\{s,a,c,t\}) = 2$
 $c(\{s,a,b,c,t\}) = 1$
 $c(\{s,b,c,t\}) = 4$
 $c(\{s,d,t\}) = 2$
 $c(\{s,d,c,t\}) = 2$
 $c(\{s,d,b,c,t\}) = 2$



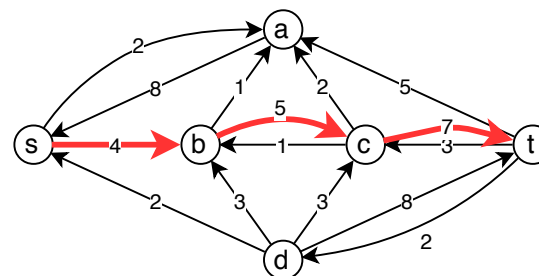
Augmenting paths:
 $c(\{s,a,c,t\}) = 2$
 $c(\{s,a,b,c,t\}) = 1$
 $c(\{s,d,t\}) = 2$
 $c(\{s,d,c,t\}) = 2$
 $c(\{s,d,b,c,t\}) = 2$



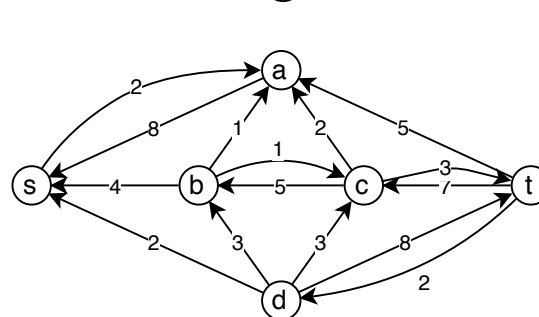
Augmenting paths:
 $c(\{s,a,b,c,t\}) = 1$
 $c(\{s,d,t\}) = 2$
 $c(\{s,d,c,t\}) = 2$
 $c(\{s,d,b,c,t\}) = 2$



Augmenting paths:
 $c(\{s,d,t\}) = 2$
 $c(\{s,d,c,t\}) = 2$
 $c(\{s,d,b,c,t\}) = 2$



Augmenting paths:
 $c(\{s,b,c,t\}) = 4$



Augmenting paths:
 None

min cut: $S = \{s,a\}$ $T = \{b,c,d,t\}$ $c(S,T) = 14$