Chapter 5

The Network Simplex Algorithm

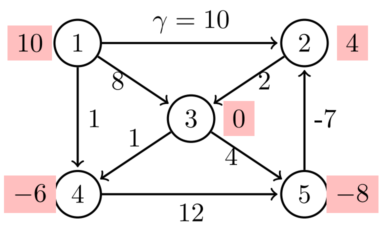
Recall the problem of min cost flow discussed in Chapter 3. We are given a network $G = (V, E)$, with $|V| = n$ nodes, $|E| = m$ edges. Each node v has a demand $d(v)$ which can be positive or negative, and a demand function d must satisfy $\sum_{v\in V} d(v) = 0$. Each edge e has a cost $\gamma(e)$, which can also be positive or negative. We recall from Definition 3.8 that a flow in such a network is a map $f : E \to \mathbb{R}$ with

1. $0 \leq f(e) \leq c(e)$ for all $e \in E$, where $c(e)$ represents the edge capacity,
2. $d(v) = \sum_{u\in O(v)} f(v, u) - \sum_{u\in I(v)} f(u, v)$ for all $v \in V$,

and that the min cost flow problem consists of finding a flow such that the cost $\gamma(f) = \sum_{e\in E} \gamma(e)f(e)$ is minimized.

For now, we assume that all capacities are infinite.

The examples and notes below are closely following the notes [2]. As an example, consider a network with $n = 5$ nodes and $m = 8$ edges. Nodes 4 and 5 are in demand ($d(v) < 0$ for $v = 4, 5$), node 3 is a transit node ($d(v) = 0$ for $v = 3$) and nodes 1 and 2 are in supply ($d(v) > 0$ for $v = 1, 2$). We can check that $\sum_{v\in V} d(v) = 10 + 4 - 8 - 6 = 0$. The cost $\gamma(e)$ is written next to each edge e.



With the notation $d_i = d(i)$, $\gamma_{ij} = \gamma((i, j))$ and $f_{ij} = f((i, j))$, write a demand vector d, a cost vector γ and the cost function $\gamma(f)$ as

$$d = \begin{bmatrix} 10 \\ 4 \\ 0 \\ -6 \\ -8 \end{bmatrix}, \ \gamma = \begin{bmatrix} \gamma_{12} \\ \gamma_{13} \\ \gamma_{14} \\ \gamma_{23} \\ \gamma_{34} \\ \gamma_{35} \\ \gamma_{45} \\ \gamma_{52} \end{bmatrix} = \begin{bmatrix} 10 \\ 8 \\ 1 \\ 2 \\ 1 \\ 4 \\ 12 \\ -7 \end{bmatrix}, \ \gamma(f) = \sum_{e \in E} \gamma(e) f(e) = \gamma^T f.$$

Thus solving the min cost flow problem in this network is equivalent to solve

$$\min \gamma^T f$$

subject to the constraints that define a flow, namely

1. $0 \leq f_{ij}$ since all edges capacities are infinite for now,
2. $d_i = \sum_j f_{ij} - \sum_j f_{ji}$ for all i.

The first constraint just says $f \geq 0$, so let us look at the second constraint. Take $i = 1$, and the demand d_1 which is 10 at node 1 must be satisfied, that is $10 = f_{12} + f_{13} + f_{14}$ must hold. Repeating this constraint for every node in the network gives:

$$\begin{array}{ll} \min & \gamma^T f \\ s.t & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} f_{12} \\ f_{13} \\ f_{14} \\ f_{23} \\ f_{34} \\ f_{35} \\ f_{45} \\ f_{52} \end{bmatrix} = \begin{bmatrix} 10 \\ 4 \\ 0 \\ -6 \\ -8 \end{bmatrix} \\ & f \geq 0, \end{array}$$

or in short,

$$\begin{array}{rl} \min & \gamma^T f \\ s.t & Af = d \\ & f \geq 0, \end{array}$$

where the matrix A is actually the *incidence matrix* of the graph G, that is A is an $n \times m$ matrix whose n rows and m columns correspond to the nodes and edges of G respectively, such that $a_{i,j} = -1$ if the edge that labels the jth column enters node i, 1 if it leaves i and 0 otherwise (the opposite sign convention is also used). In our example, row 2 corresponds to node 2, the columns are indexed by the edges:

$$\begin{array}{cccccccc} (1,2) & (1,3) & (1,4) & (2,3) & (3,4) & (3,5) & (4,5) & (5,2) \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{array}$$

and the edge $(1,2)$ enters 2, so the sign is -1, while $(2,3)$ leaves 2, so the sign is 1. Columnwise, the column for (i,j) will have a 1 in row i and a -1 in row j.

Notice that the rows of A are linearly dependent, if you sum them, you obtain the whole zero vector. The linear dependency is happening because of the constraint that $\sum_{v\in V} d(v) = 0$, and the positive demand must be equal to the negative one. This means that one of the equations can be canceled.

5.1 Uncapacitated Networks

In min cost flow networks, the flow f is the unknown to be found, so we will change the notation and use x to denote the flow, this makes the notation consistent with linear programming, the view point we will adopt next, without hopefully creating confusion. We have motivated above that the min cost flow network problem in the case where the edge capacities are unbounded (in which case the network is called uncapacitated) can be formulated as a linear program:

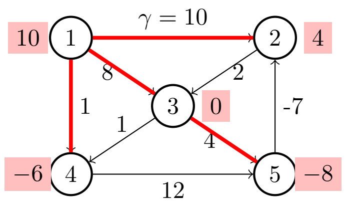
$$\begin{array}{rl} \min & \gamma^T x \\ s.t & Ax = d \\ & x \geq 0, \end{array}$$

where the $n \times m$ matrix A is the incidence matrix of the graph G. The rank of A is $n-1$ (one row equation is redundant, because positive demand must equal negative demand), thus A contains $n-1$ linearly independent columns.

Example 5.1. Consider again the network of the introduction with incidence matrix

$$\begin{array}{cccccccc} (1,2) & (1,3) & (1,4) & (2,3) & (3,4) & (3,5) & (4,5) & (5,2) \end{array}$$
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix}.$$

Then the columns corresponding to $(1,2)(1,3)(1,4)(3,5)$ are linearly independent, and they form a spanning tree (disregarding orientation):



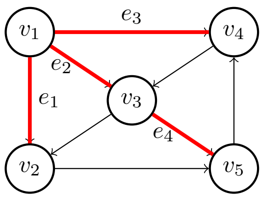
$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

We will show that this is no coincidence, that that for a connected network, $n-1$ linearly independent vectors correspond to a spanning tree (disregarding orientation), and vice-versa.

Lemma 5.1. *Let T be a set of columns of A such that the corresponding edges form a spanning tree. Then the columns in T are linearly independent.*

Proof. Every tree is made of n nodes $v_1, \ldots, v_n$, and for every $i \geq 2$, there is exactly one edge with end point equal to v_i, and the other equal to one of the previously labeled $v_1, \ldots, v_{i-1}$. So visit the tree and label the nodes by $v_1, \ldots, v_n$ and the edges by $e_1, \ldots, e_{n-1}$ in order of visit. Then order the $n-1$ columns of the spanning tree and the n rows of A following the same ordering. Discard the first row, to obtain an $(n-1) \times (n-1)$ matrix A' which is upper triangular and contains nonzero elements on its diagonal by construction. Its determinant is the product of the diagonal coefficients, thus $\det(A') \neq 0$ and A has rank $n-1$, showing that the columns in T are linearly independent. □

Example 5.2. We illustrate the algorithm given in this proof on our running example. To start with, if we visit the nodes in the order of their labels, that is, $v_i = i$, $i = 1, \ldots, 5$, then the edges are visited in the order $e_1 = (1,2)$, $e_2 = (1,3)$, $e_3 = (1,4)$, $e_4 = (3,5)$, which is the current ordering for the columns of the rank $n-1$ submatrix which is already such that once the first row is removed, it is an upper diagonal matrix. But we can also change the ordering, e.g., choose $v_1 = 1$, $v_2 = 4$, $v_3 = 3$, $v_4 = 2$, $v_5 = 5$, then edges are visited in the order $e_1 = (v_1, v_2)$, $e_2 = (v_1, v_3)$, $e_3 = (v_1, v_4)$, $e_4 = (v_3, v_5)$.



$$\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{bmatrix} 1 & 1 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

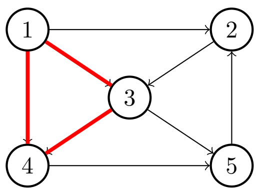
Conversely:

Lemma 5.2. *If a subset T of $n-1$ columns of A are linearly independent, then the corresponding edge set is a spanning tree.*

Proof. We do a proof by contradiction. Suppose that the corresponding edge set is not a spanning tree, then it must contain a cycle (we disregard the orientation of the edges for the cycle). It is sufficient to show that the columns of A associated with a cycle are linearly dependent.

Given a cycle, the coefficients of the linear combination are obtained as follows. Arbitrarily choose one direction for the cycle, and set the coefficient of the column of (i,j) in the cycle as 1 if the edge has the same orientation as the cycle, and -1 otherwise. This linear combination of the columns of A involved in the cycle yields the zero vector. □

Example 5.3. Consider the cycle $(1,3),(3,4),(4,1)$. Then the sign for the column of $(1,3)$ should be 1, the sign for the column of $(3,4)$ should be 1 as well, and the sign for the column of $(1,4)$ should be -1.



$$\begin{array}{ccc} (1,3) & (3,4) & (1,4) \\ \end{array}$$
$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

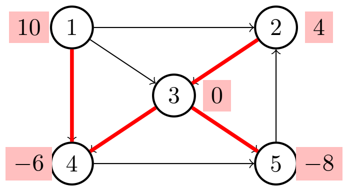
Both lemmas together are summarized in this theorem.

Theorem 5.3. *Given a connected flow network, with incidence matrix A, a submatrix A_T of size $(n-1)\times(n-1)$ has rank $n-1$ if and only if the edges associated with the set T of columns of A_T form a spanning tree.*

Note that this statement considers a square submatrix A_T, this is because once $n-1$ columns are chosen which correspond to a spanning tree, the incidence matrix A has n rows, thus we start by by considering an $n\times(n-1)$ matrix of rank $n-1$, out of which we get A_T by removing a row.

Basic Feasible Solutions. We now recall that a basic solution for a linear program (LP) is obtained by choosing $n-1$ linearly independent columns of the constraint matrix A (where $n-1$ is the rank of A). This means, a basic solution for the min cost flow problem is a spanning tree. As for any LP, when taking $n-1$ columns of A, we can get columns which are dependent, but when they are not (that is, we have a spanning tree), let A_T denote the corresponding matrix (with one row removed, again, one row is redundant), and we have either $x = A_T^{-1}d \geq 0$, that is x is a basic feasible solution (BFS), or $x = A_T^{-1}d < 0$ (this is a basic solution, but it is infeasible). Presence of a basic variable taking value 0 gives a degenerate solution (that is, we have an empty edge in the spanning tree, a *degenerate spanning tree*), as for standard LPs.

Example 5.4. As a first example, consider the spanning tree showed below and its corresponding columns of A.



$$\begin{array}{cccc} (1,4) & (3,4) & (2,3) & (3,5) \\ \end{array}$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & -1 & 1 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

We need to solve

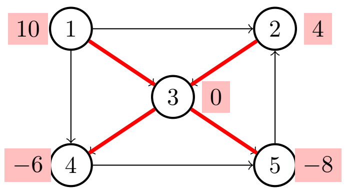
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & -1 & 1 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_{14} \\ x_{34} \\ x_{23} \\ x_{35} \end{bmatrix} = \begin{bmatrix} 10 \\ 4 \\ 0 \\ -6 \\ -8 \end{bmatrix}.$$

We could remove a row of this matrix to get A_T and invert it, but it is probably easier to just solve the system, since we immediately see that:

$$x_{14} = 10, x_{23} = 4, x_{35} = 8, -x_{14} - x_{34} = -6 \Rightarrow x_{34} = -4.$$

This solution is thus basic but not feasible since $x_{34} = -4$.

As a second example, consider the spanning tree showed below and its corresponding columns of A.



$$\begin{array}{cccc} (1,3) & (2,3) & (3,4) & (3,5) \\ \end{array}$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -1 & 1 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

We need to solve

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -1 & 1 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_{13} \\ x_{23} \\ x_{34} \\ x_{35} \end{bmatrix} = \begin{bmatrix} 10 \\ 4 \\ 0 \\ -6 \\ -8 \end{bmatrix}.$$

We immediately see that:

$$x_{13} = 10, x_{23} = 4, x_{34} = 6, x_{35} = 8$$

and we get a BFS.

Checking for Optimality. Now that we know how to get basic feasible solutions (BFS) for our min cost flow problem, the next step is to see whether it is optimal. We will use duality theory for that.

The min cost flow problem is:

$$\begin{aligned} \min \quad & \gamma^T x \\ s.t \quad & Ax = d \\ & x \geq 0, \end{aligned}$$

which is equivalent to

$$\begin{aligned} \max \quad & -\gamma^T x \\ s.t \quad & Ax \leq d \\ & Ax \geq d \\ & x \geq 0, \end{aligned}$$

and we can alternatively write

$$\begin{aligned} (P) : \max \quad & -\gamma^T x \\ s.t \quad & \begin{bmatrix} A \\ -A \end{bmatrix} x \leq \begin{bmatrix} d \\ -d \end{bmatrix} \\ & x \geq 0, \end{aligned}$$

where the constraint matrix is of size $2n \times m$, and x is the vector containing the flow on all the m edges of the network.

The dual problem is thus given by:

$$\begin{aligned} \min \quad & [d^T, -d^T] \begin{bmatrix} s \\ t \end{bmatrix} \\ s.t \quad & \begin{bmatrix} s^T & t^T \end{bmatrix} \begin{bmatrix} A \\ -A \end{bmatrix} \geq -\gamma^T \\ & s, t \geq 0. \end{aligned}$$

Let us rewrite a bit the dual:

$$\begin{aligned} \min \quad & d^T s - d^T t \\ s.t \quad & s^T A - t^T A \geq -\gamma^T \\ & s, t \geq 0 \\ \Rightarrow \min \quad & d^T (s - t) \\ s.t \quad & A^T (s - t) \geq -\gamma \\ & s, t \geq 0 \\ \Rightarrow \max \quad & d^T u \\ s.t. \quad & -A^T u \geq -\gamma \end{aligned}$$

by setting $u := t - s$. Note that $t, s \geq 0$ means that u can be positive or negative. This gives our final dual:

$$\begin{aligned} (D) : \max \quad & d^T u \\ s.t \quad & A^T u \leq \gamma \end{aligned}$$

or equivalently, since we recall that in the transpose A^T of the incidence matrix A, every row contains exactly one 1 in column i and one -1 in column j, for the edge (i, j):

$$\begin{aligned} (D) : \max \quad & \textstyle\sum_{i=1}^n d_i u_i \\ s.t \quad & u_i - u_j \leq \gamma_{ij}, \text{for all } (i, j) \in E. \end{aligned}$$

Let us keep in mind that the primal has one redundant equation, which means that the dual has one redundant variable, which can be set to zero without altering the problem.

Consider a basic feasible solution for the primal, and partition the edges of the network into the set B of edges of the spanning tree, and the set N of other edges (since we know that the edges of the spanning tree corresponding to a basic solution, we use the LP notation B and N).

If an edge (i, j) in B is in the optimal solution (which is not degenerate), then $x_{ij} > 0$ and by complementary slackness from Theorem 4.10, the corresponding dual constraint must be satisfied with equality:

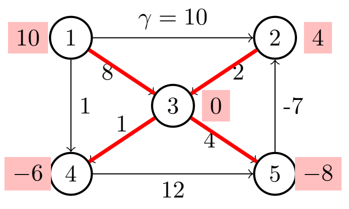
$$u_i - u_j = \gamma_{ij}, \text{ for all } (i, j) \in B$$

for u_i, u_j coefficients of an optimal solution u for the dual. There are $n-1$ such equations, one for each edge of the spanning tree, and we have n variables, one of them being zero (corresponding to the redundant equation in A). With a system of $n-1$ equations in $n-1$ variables, we can solve for u, and check that u is feasible (otherwise it cannot be optimal), namely check that

$$u_i - u_j \leq \gamma_{ij}, \text{ for all } (i, j) \in N.$$

If u satisfies the complementary slackness conditions, it is optimal by the Equilibrium Theorem (Theorem 4.10) (namely, both x and u are feasible, (2) is satisfied by construction and (1) is always satisfied since all constraints for the primal are equalities).

Example 5.5. Consider the spanning tree showed below



$$\begin{array}{cccc} (1,3) & (2,3) & (3,4) & (3,5) \end{array}$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -1 & 1 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_{13} \\ x_{23} \\ x_{34} \\ x_{35} \end{bmatrix} = \begin{bmatrix} 10 \\ 4 \\ 0 \\ -6 \\ -8 \end{bmatrix}.$$

and its corresponding BFS:

$$x_{13} = 10, x_{23} = 4, x_{34} = 6, x_{35} = 8.$$

The corresponding dual equations are:

$$\begin{aligned} u_1 - u_3 &= \gamma_{13} = 8 \\ u_2 - u_3 &= \gamma_{23} = 2 \\ u_3 - u_4 &= \gamma_{34} = 1 \\ u_3 - u_5 &= \gamma_{35} = 4 \end{aligned}$$

We choose to set $u_3 = 0$ (any u_i could be chosen, but u_3 appears often, so it suggests that it will be easy to solve the system). Then

$$u_1 = 8,\ u_2 = 2,\ u_4 = -1,\ u_5 = -4.$$

Finally we check the feasibility of u by looking at $u_i - u_j \leq \gamma_{ij}$ for $(i,j) \in N$:

$$\begin{array}{lll}
u_1 - u_2 & \leq \gamma_{12} = 10 \Rightarrow 8 - 2 \leq 10 & \checkmark \\
u_1 - u_4 & \leq \gamma_{14} = 1 \Rightarrow 8 + 1 = 9 \not\leq 1 & \times \\
u_4 - u_5 & \leq \gamma_{45} = 12 & \\
u_5 - u_2 & \leq \gamma_{52} = -7 &
\end{array}$$

so $u = (8, 2, 0, -1, -4)$ is not optimal.

If we keep in mind the structure of the general simplex algorithm (see Algorithm 10) which we recall below for convenience, what we have done so far is start with an initial BFS (described in terms of spanning tree), then check for optimality (by complementary slackness).

Algorithm 10 General Simplex Algorithm

```
    Input: an LP in standard form max $c^T x$, such that $Ax = b,\ x \geq 0$.
    Output: a vector $x^*$ that maximizes the objective function $c^T x$.
1: Start with an initial BFS.
2: while (the current BFS is not optimal) do
3:     Move to an improved adjacent BFS.
4: return $x^*$ =BFS;
```

We are now left with moving to an improved adjacent BFS (or verify that the problem is unbounded).

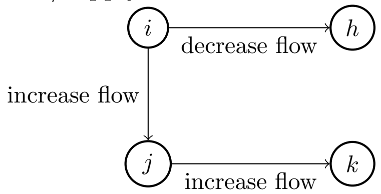
Pivoting. If a solution x does not verify the optimality conditions, there must exist an edge $(i,j) \in N$ such that

$$u_i - u_j > \gamma_{ij} \iff \gamma_{ij} - u_i + u_j = \gamma_{ij} - u^T A_{ij} < 0$$

where A_{ij} denotes the column of A associated to (i,j). We have seen in the proof of the Strong Duality Theorem (see (4.2)) that the reduced cost associated to x is given by $\gamma - A^T u$ thus $\gamma_{ij} - u_i + u_j$ is the reduced cost associated to the variable x_{ij}, and having a positive reduced cost is profitable when it comes to minimize the cost of the flow. We thus need to update u_i, u_j such that they satisfy this inequality instead of violating it, but then, this means that x_{ij} will change accordingly, and will go from being 0 to nonzero. This means activating the edge (i,j). We recognize the pivoting phase of the simplex algorithm, where a variable x_{ij} moves from N to B.

So add (i,j) to B. Then (i,j) forms a cycle within the spanning tree. In order to restore a BFS, we must get rid of this cycle, that is remove one edge of the cycle. Let us see which one.

To maintain the feasibility of the current solution, one must necessarily alter the value of the flow on all edges of the cycle, increasing the flow on edges that have the same orientation as (i, j) (called direct edges), and decreasing the flow of the edges having opposite direction (reverse edges), to keep the demand/supply satisfied.



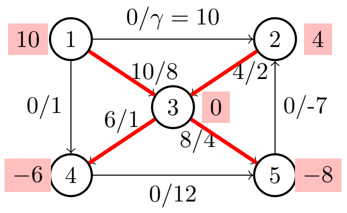
- If there is at least one reverse edge, as the flow on (i, j) is increased, the flow on reverse edges decreases, and there is at least one edge, say (u, v) such that its flow reaches zero before all the other edges, then (u, v) leaves B. The maximum feasible value ϑ of the flow in (i, j) is then

  $$\vartheta(i, j) = \min\{x_{hk},\ (h, k) \text{ reverse edge in the cycle}\}.$$

  The new BFS is thus obtained by simply increasing by ϑ the flow in direct edges and decreasing by the same amount the flow in reverse edges of the cycle.

- If all edges are direct, then we know that increasing the flow on (i, j) reduces the cost, so we increase the flow on the other edges of the cycle accordingly, and since we consider an uncapacitated network, there is no upper bound on how much we can increase the flow on (i, j) (or on any edge of the cycle) and the problem is then unbounded. Note that we are activating the edge (i, j) because it improves the objective function, so the only way the cost is getting minimized by increasing the flow on every edge is when $\sum_{(h,k) \text{ in cycle}} \gamma_{hk} < 0$ (this is a negative cycle of infinite capacity, see Definition 3.10 and the remarks below).

Example 5.6. We continue the example given by the spanning tree below, whose corresponding BFS is the flow shown:



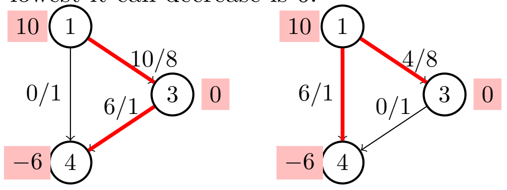
We check the feasibility of u by looking at $u_i - u_j \leq \gamma_{ij}$ for $(i, j) \in N$ and we already found in the previous example an inequality that is not satisfied:

$$\begin{aligned} u_1 - u_2 &= \gamma_{12} = 10 \Rightarrow 8 - 2 \leq 10 \ \checkmark \\ u_1 - u_4 &= \gamma_{14} = 1 \Rightarrow 8 + 1 = 9 \not\leq 1 \ \times \end{aligned}$$

We thus activate the edge (1,4), which creates a cycle $(1) \to (3) \to (4) \leftarrow (1)$, and

$$\vartheta(1,4) = \min\{x_{13}, x_{34}\} = \min\{6, 10\} = 6.$$

When we are pushing more flow on (1,4), it decreases on the reverse edges, and the lowest it can decrease is 6.



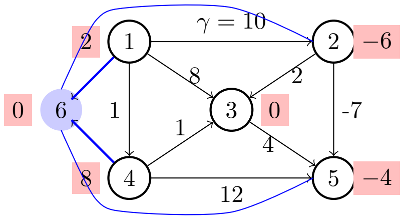
We thus update the BFS:

$$\begin{bmatrix} x_{12} \\ x_{13} \\ x_{14} \\ x_{23} \\ x_{34} \\ x_{35} \\ x_{45} \\ x_{52} \end{bmatrix} = \begin{bmatrix} 0 \\ 10 \\ 0 \\ 4 \\ 6 \\ 8 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} x_{12} \\ x_{13} \\ x_{14} \\ x_{23} \\ x_{34} \\ x_{35} \\ x_{45} \\ x_{52} \end{bmatrix} = \begin{bmatrix} 0 \\ 4 \\ 6 \\ 4 \\ 0 \\ 8 \\ 0 \\ 0 \end{bmatrix}.$$

We recognize the pivot operation, where x_{14} goes out of B, and is replaced by x_{34}. See Exercise 45 for the end of this example.

Artificial variables. Both the network simplex algorithm and the simplex algorithm start with a basic feasible solution. In the regular simplex algorithm, we saw that we can use artificial variables to find a basic feasible solution that will serve as a starting point for the algorithm.

Let us mimic this for the network simplex algorithm, and introduce "artificial edges". We convey all the flows produced by nodes with positive demands into an artificial node via artificial edges, and redistribute such a flow via other artificial edges, connecting the new node to nodes with a negative demand.

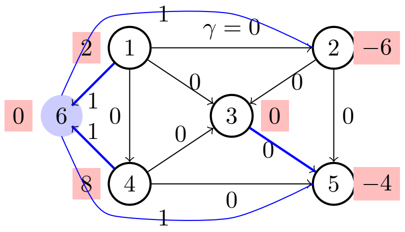


As for the simplex algorithm, the objective is to minimize the flow on artificial edges, so in the artificial problem, we set the cost of each artificial edge equal to 1, and the cost of all others to 0.

We can solve this new min cost flow problem using the network simplex algorithm, but for this to be helping with our original problem, we need to be able to find an immediate BFS. If there is no transit node, then artificial edges form an initial BFS. If there is a transit node, artificial edges are not enough to form a spanning tree of the artificial network (as illustrated above). To obtain an initial BFS, we can add an arbitrary edge of the original network for each transit node to connect it to the tree formed by the artificial edges, so that no loop is formed in the process.

If some artificial variables have a nonzero flow, the problem is infeasible. If the flow in all artificial variables is 0, these can be removed (together with artificial nodes) to get a BFS for the original network.

Example 5.7. In this network (close to our running example, but for the direction of the edges (3,4), (5,2) and demands), we created artificial edges, and added the edge (3,5) to form a spanning tree, our initial BFS. Edges in the original network have a cost of 0 (we can use them as much as we want), while artificial edges have a cost of 1, since we ideally would like not to use them.



We want to minimize $\gamma^T x$ where γ is a vector with zero everywhere, but for $\gamma_{46} = \gamma_{16} = \gamma_{62} = \gamma_{65} = 1$. We have an immediate BFS given by (note the last equation which describes that the artificial node 6 is a transit node)

$$\begin{array}{c} (1,6)(3,5)(4,6)(6,2)(6,5) \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 \\ -1 & 0 & -1 & 1 & 1 \end{bmatrix} \end{array} \begin{bmatrix} x_{16} \\ x_{35} \\ x_{46} \\ x_{62} \\ x_{65} \end{bmatrix} = \begin{bmatrix} 2 \\ -6 \\ 0 \\ 8 \\ -4 \\ 0 \end{bmatrix}$$

from which we get

$$\begin{bmatrix} x_{16} \\ x_{35} \\ x_{46} \\ x_{62} \\ x_{65} \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 8 \\ 6 \\ 4 \end{bmatrix}.$$

In words, we are just setting the flow on the artificial edges so that it goes from the supply nodes to the demand nodes, based on their respective demands.

Using complementary slackness, we get:

$$\begin{aligned} u_1 - u_6 &= \gamma_{16} = 1 \\ u_4 - u_6 &= \gamma_{46} = 1 \\ u_6 - u_2 &= \gamma_{62} = 1 \\ u_6 - u_5 &= \gamma_{65} = 1 \\ u_3 - u_5 &= \gamma_{35} \le 0 \end{aligned}$$

and setting $u_6 = 0$ gives $u_1 = u_4 = 1$, $u_2 = u_5 - 1$, $u_3 \le -1$. Note that we have a degenerate solution since $x_{35} = 0$. We then check for feasibility:

$$\begin{aligned} u_1 - u_2 &= 2 \not\le 0 \quad \times \\ u_4 - u_3 &\ge 2 \not\le 0 \quad \times \end{aligned}$$

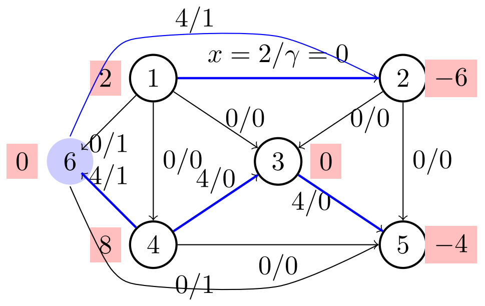
so $(1,2)$ enters B, creating the cycle $(6) \leftarrow (1) \rightarrow (2) \leftarrow (6)$, where $x_{16} = 2 < x_{62} = 6$ so $(1,6)$ leaves B. The BFS is updated, x_{16} is replaced by x_{12}, and x_{62} is decreased by 2:

$$\begin{bmatrix} x_{12} \\ x_{35} \\ x_{46} \\ x_{62} \\ x_{65} \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 8 \\ 4 \\ 4 \end{bmatrix}.$$

We notice that this cycle does not contain $(4,3)$, so the second inequality is not changing, and $(4,3)$ enters B, creating a cycle $(6) \leftarrow (4) \rightarrow (3) \rightarrow (5) \leftarrow (6)$, so $x_{65} = 4 < x_{46} = 8$ and $(6,5)$ leaves B. The BFS is updated, x_{65} is replaced by x_{43}, x_{46} is decreased by 4, and x_{35} is increased by 4:

$$\begin{bmatrix} x_{12} \\ x_{35} \\ x_{46} \\ x_{62} \\ x_{43} \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 4 \\ 4 \\ 4 \end{bmatrix}.$$

At this point, the network flow is as follows:



We use complementary slackness once more:

$$\begin{aligned}
u_1 - u_2 &= \gamma_{12} = 0\\
u_4 - u_6 &= \gamma_{46} = 1\\
u_6 - u_2 &= \gamma_{62} = 1\\
u_4 - u_3 &= \gamma_{43} = 0\\
u_3 - u_5 &= \gamma_{35} = 0
\end{aligned}$$

and setting $u_6 = 0$ gives $u_3 = u_4 = u_5 = 1$, $u_2 = u_1 = -1$. We then check for feasibility:

$$\begin{aligned}
u_1 - u_6 &= -1\\
u_1 - u_4 &= 0\\
u_1 - u_3 &= 0\\
u_4 - u_5 &= 0\\
u_6 - u_5 &= -1\\
u_2 - u_5 &= -2
\end{aligned}$$

and since γ_{ij} takes for value 0 and 1, the vector u is feasible. Thus we have an optimal solution, but two artificial edges are not set to 0, thus the original problem is infeasible.

In fact, this is something that can be easily checked from the original network: node 2 is unreachable from node 4, and node 1 is unable to satisfy the demand of node 2.

5.2 Capacitated Networks

Suppose now that the edges have a finite capacity, that is the for every edge $e \in E$, $f(e) \leq c(e)$. We need to modify the current LP program

$$\begin{aligned}
\min \quad & \gamma^T x\\
s.t \quad & Ax = d\\
& x \geq 0
\end{aligned}$$

and add the capacity constraint $x \leq c$ where c is the vector containing the capacities. Introducing a slack vector s, this becomes

$$\begin{aligned}
\min \quad & \gamma^T x\\
s.t \quad & \begin{bmatrix} A & 0_m \\ I_m & I_m \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} = \begin{bmatrix} d \\ c \end{bmatrix}\\
& x, s \geq 0.
\end{aligned}$$

Basic Feasible Solutions. We saw that in the uncapacitated case, basic solutions correspond to spanning trees, so we could partition the edges into the set B of edges belonging to the spanning tree, and the set N of the other edges.

For capacitated networks, we partition the edges into three sets, the set $\mathcal{S}$ of saturated edges, that is edges where the flow is reaching its capacity, the set $\mathcal{N}$ of edges with zero flow , and the set $\mathcal{B}$ containing the other edges. We show next how this tripartition will help us to find basic feasible solutions.

Theorem 5.4. *Given a connected flow network, a basic feasible solution x and the corresponding tripartition $(\mathcal{N}, \mathcal{S}, \mathcal{B})$ of the edges, the edges in $\mathcal{B}$ do not form cycles.*

Proof. Suppose there is a cycle (disregarding direction) in $\mathcal{B}$, say (i_1, i_2),(i_2, i_3), (i_4, i_3), (i_4, i_5), (i_1, i_5), fix the cycle direction to be that of (i_1, i_2), thus (i_1, i_2), (i_2, i_3), (i_4, i_5) are in the same direction, and (i_4, i_3), (i_1, i_5) are not.

Let us look at the corresponding columns in the constraint matrix given by:

$$\begin{bmatrix} A & 0_m \\ I_m & I_m \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} = \begin{bmatrix} d \\ c \end{bmatrix}$$

Since the cycle is formed by edges in $\mathcal{B}$, this means that none of these edges are saturated, and so to each of them correspond a non-zero slack variable. The columns (and rows) of the constraint matrix involving the cycle are thus:

$$\begin{array}{c|cccccccccc|}
 & x_{i_1,i_2} & x_{i_2,i_3} & x_{i_4,i_3} & x_{i_4,i_5} & x_{i_1,i_5} & s_{i_1,i_2} & s_{i_2,i_3} & s_{i_4,i_3} & s_{i_4,i_5} & s_{i_1,i_5} \\
i_1 & 1 & 0 & 0 & 0 & 1 & & & & & \\
i_2 & -1 & 1 & 0 & 0 & 0 & & & & & \\
i_3 & 0 & -1 & -1 & 0 & 0 & & & & & \\
i_4 & 0 & 0 & 1 & 1 & 0 & & & & & \\
i_5 & 0 & 0 & 0 & -1 & -1 & & & & & \\
(i_1,i_2) & 1 & & & & & 1 & & & & \\
(i_2,i_3) & & 1 & & & & & 1 & & & \\
(i_4,i_3) & & & 1 & & & & & 1 & & \\
(i_4,i_5) & & & & 1 & & & & & 1 & \\
(i_1,i_5) & & & & & 1 & & & & & 1
\end{array}$$

The upper part can be interpreted as usual, in terms of edges. The lower part represents slack variables equations, (i_1, i_2) means that s_{i_1,i_2} is the slack variable corresponding to x_{i_1,i_2}.

Since x is a BFS, and the columns of $x_{i_1,i_2}, x_{i_2,i_3}, x_{i_4,i_5}$ are linearly independent, $x_{i_1,i_2}, x_{i_2,i_3}, x_{i_4,i_5}$ are part of the basic variables. So are the slack variables $s_{i_1,i_2}, s_{i_2,i_3}, s_{i_4,i_5}$, since their columns are linearly independent as well.

Now perform the following linear combination of the columns: use $(+1)$ for the columns of $x_{i,j}$ with direct edges and of $s_{i,j}$ with reverse edges, and use (-1) for the columns of $x_{i,j}$ with reverse edges, and $s_{i,j}$ with direct edges.

The choice of the coefficients ± 1 for $x_{i,j}$ makes explicit the presence of a cycle (and thus a linear combination), as done in the proof of Lemma 5.2.

$$\begin{array}{c} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ (i_1,i_2) \\ (i_2,i_3) \\ (i_4,i_3) \\ (i_4,i_5) \\ (i_1,i_5) \end{array} \begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & +1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & +1 \\ 1 & & & & \\ & 1 & & & \\ & & -1 & & \\ & & & 1 & \\ & & & & -1 \end{bmatrix}$$

Thus when summing these columns, the upper part becomes a zero vector. The choices of ± 1 for $s_{i,j}$ ensures that the lower part also becomes a zero vector once the remaining columns are added. This shows that we obtain a zero vector, the columns are thus linearly dependent, and therefore, the corresponding 10 variables cannot all be basic variables, so some of the $x_{i_4,i_3}, x_{i_1,i_5}, s_{i_4,i_3}, s_{i_1,i_5}$ are not and must be zero. If either x_{i_4,i_3} or x_{i_1,i_5} is 0, then its edge belongs to $\mathcal{N}$ and not to $\mathcal{B}$, and if either s_{i_4,i_3} or s_{i_1,i_5} is 0, then its edge belongs to $\mathcal{S}$ and not to $\mathcal{B}$, a contradiction in all cases, and thus there cannot be a cycle in $\mathcal{B}$. □

We have a network of n nodes and the largest set of edges we can have with no cycle is a spanning tree, which involves n nodes and $n-1$ edges.

As a consequence of the above theorem:

- In a basic solution, at most $n-1$ edges can belong to $\mathcal{B}$ (because adding one more edge necessarily gives a cycle).

Thus a set $\mathcal{B}$ with $n-1$ edges correspond to a spanning tree. However, unlike in the uncapacitated case, where the spanning tree gives immediately a basic feasible solution, in this case, we need to know $\mathcal{S}$ and $\mathcal{N}$, based on which we get a system of linear equations from which we can compute the BFS (so the BFS does not only depend on the spanning tree, but also on $\mathcal{S}$ and $\mathcal{N}$).

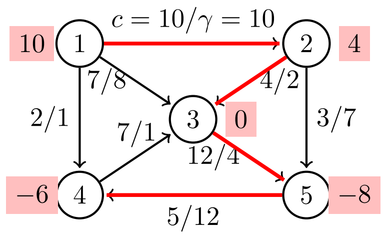
Note that:

- A choice of $\mathcal{S}$ and $\mathcal{N}$ means that we are setting flows with edges in $\mathcal{N}$ to be zero (non-basic variables), and flows with edges in $\mathcal{S}$ to be equal to their capacities (that is the corresponding slack variables become 0, thus are non-basic variables). So this sets basic and non-basic variables.

- We are then left with exactly $n-1$ basic variables to be computed, and they correspond to the edges of the spanning tree. We use demands at the nodes to obtain $n-1$ equations, so we can solve the system. We obtain a basic solution, but we need to check whether it is a basic feasible solution.

If there are saturated or empty edges in $\mathcal{B}$, then we have a degenerate basic solution (and a degenerate spanning tree).

Example 5.8. Consider the network below, where every edge e is given a capacity $c(e)$ and a cost $\gamma(e)$ (the lower capacity is assumed to be 0).



We can write a demand vector d, a cost vector γ and a capacity vector c as

$$d = \begin{bmatrix}10\\4\\0\\-6\\-8\end{bmatrix}, \ \gamma = \begin{bmatrix}\gamma_{12}\\\gamma_{13}\\\gamma_{14}\\\gamma_{23}\\\gamma_{34}\\\gamma_{35}\\\gamma_{45}\\\gamma_{52}\end{bmatrix} = \begin{bmatrix}10\\8\\1\\2\\7\\4\\1\\12\end{bmatrix}, \ c = \begin{bmatrix}c_{12}\\c_{13}\\c_{14}\\c_{23}\\c_{34}\\c_{35}\\c_{45}\\c_{52}\end{bmatrix} = \begin{bmatrix}10\\7\\2\\4\\3\\12\\7\\5\end{bmatrix}.$$

Consider the spanning tree given by $\mathcal{B} = \{(1,2),(2,3),(3,5),(5,4)\}$. Suppose that $\mathcal{N} = \{(4,3)\}$, $\mathcal{S} = \{(1,3),(1,4),(2,5)\}$.

The edge $(4,3) \in \mathcal{N}$ which means its flow is 0, so $x_{43} = 0$. For the edges in $\mathcal{S}$, they are saturated, that means:

$$x_{13} = c_{13} = 7, \ x_{14} = c_{14} = 2, \ x_{25} = c_{25} = 3.$$

So we have $n - 1 = 4$ flow variables to be computed for edges in $\mathcal{B}$, and we already know the values of the 4 flow variables given by $\mathcal{N}$ and $\mathcal{S}$, so we use demands at the nodes to solve:

$$\begin{aligned}
x_{12} &= d_1 - x_{14} - x_{13} = 10 - 2 - 7 = 1\\
x_{23} &= d_2 + x_{12} - x_{25} = 4 + 1 - 3 = 2\\
x_{35} &= d_3 + x_{13} + x_{23} = 7 + 2 = 9\\
x_{54} &= d_4 + x_{35} + x_{25} = -8 + 3 + 9 = 4,
\end{aligned}$$

and we get a basic solution which is feasible.

Optimality. To figure out whether a BFS is optimal, we use duality, as for the uncapacitated case.

To the primal program:

$$\begin{aligned}
(P) : \min \quad & \gamma^T x\\
s.t \quad & \begin{bmatrix}A & 0_m\\ I_m & I_m\end{bmatrix}\begin{bmatrix}x\\s\end{bmatrix} = \begin{bmatrix}d\\c\end{bmatrix}\\
& x, s \geq 0
\end{aligned}$$

corresponds the dual program (see Exercise 46 for the details of the computation):

$$\begin{aligned} \max \quad & [d^T, c^T] \begin{bmatrix} u \\ v \end{bmatrix} \\ s.t \quad & \begin{bmatrix} A^T & I_m \\ 0_m & I_m \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \leq \begin{bmatrix} \gamma \\ 0_m \end{bmatrix}. \end{aligned}$$

We then invoke complementary slackness. The primal involves the variables x and s. If $(i,j) \in \mathcal{B}$, then $x_{ij} > 0$ and $s_{ij} > 0$ (since the edge is not saturated), and complementary slackness for x_{ij} gives $u_i - u_j + v_{ij} = \gamma_{ij}$, while complementary slackness for s_{ij} gives $v_{ij} = 0$, and $u_i - u_j = \gamma_{ij}$:

$$\gamma_{ij} - u_i + u_j = 0, \ (i,j) \in \mathcal{B}.$$

As for the uncapacitated case, we get a system of $n-1$ equations in n variables, one of which can be fixed to 0. The values for the coefficients of u that are computed must satisfy the other dual constraints.

For an empty edge $(i,j) \in \mathcal{N}$, complementary slackness for $s_{ij} > 0$ gives $v_{ij} = 0$, so with $u_i - u_j + v_{ij} \leq \gamma_{ij}$:

$$u_i - u_j \leq \gamma_{ij}, \ (i,j) \in \mathcal{N}.$$

For a saturated edge, $x_{ij} = c_{ij} > 0$ (an edge with 0 capacity can just be ignored in the network), complementary slackness for x_{ij} gives $u_i - u_j = \gamma_{ij} - v_{ij}$, with $v_{ij} \leq 0$, and we get:

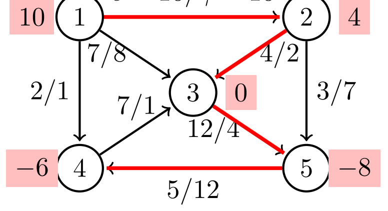
$$u_i - u_j \geq \gamma_{ij}, \ (i,j) \in \mathcal{S}.$$

If all equations

$$\gamma_{ij} - u_i + u_j = 0, \ (i,j) \in \mathcal{B}, \ u_i - u_j \leq \gamma_{ij}, \ (i,j) \in \mathcal{N}, \ u_i - u_j \geq \gamma_{ij}, \ (i,j) \in \mathcal{S},$$

are satisfied, then the flow is optimal.

Example 5.9. We continue Example 5.8, with spanning tree given by $\mathcal{B} = \{(1,2),(2,3),(3,5),(5,4)\}$, $\mathcal{N} = \{(4,3)\}$, and $\mathcal{S} = \{(1,3),(1,4),(2,5)\}$, corresponding to the BFS $x_{12} = 1$, $x_{23} = 2$, $x_{35} = 9$, $x_{54} = 4$ with $x_{13} = c_{13} = 7$, $x_{14} = c_{14} = 2$, $x_{25} = c_{25} = 3$ and $x_{43} = 0$.



For all $(i,j) \in \mathcal{B}$, $\gamma_{ij} - u_i + u_j = 0$:

$$\begin{aligned}
\gamma_{12} - u_1 + u_2 = 0 &\quad \Rightarrow 10 - u_1 + u_2 = 0\\
\gamma_{23} - u_2 + u_3 = 0 &\quad \Rightarrow 2 - u_2 + u_3 = 0\\
\gamma_{35} - u_3 + u_5 = 0 &\quad \Rightarrow 4 - u_3 + u_5 = 0\\
\gamma_{54} - u_5 + u_4 = 0 &\quad \Rightarrow 12 - u_5 + u_4 = 0.
\end{aligned}$$

Set $u_4 = 0$. Then $u_5 = 12$, $u_3 = 16$, $u_2 = 18$, $u_1 = 28$. The vector u is not feasible, since $(2,5) \in \mathcal{S}$, but $u_2 - u_5 \geq \gamma_{25} = 7$ should be satisfied, while $18 - 12 = 4 \leq 7$.

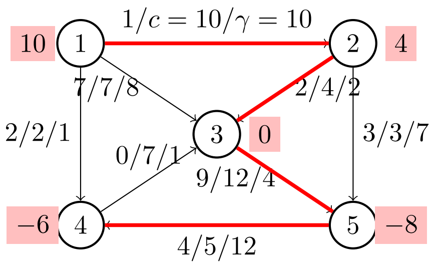
Pivoting. If the vector u is not feasible, and thus the corresponding flow x is not optimal, an edge (empty or full) that does not satisfy its constraint may be chosen to join $\mathcal{B}$, creating a cycle C, then accordingly one edge must leave.

Thus one must find the maximum flow ϑ circulating in the cycle C generated by the addition of (i,j). The value of ϑ may be limited by the capacity of the direct edges which saturate, as well as by reverse edges that are getting empty:

$$\vartheta = \min\{x_{pq},\ (p,q) \text{ direct edge in } C,\ c_{pq} - x_{pq},\ (p,q) \text{ reverse edge in } C\}.$$

The edge that determines ϑ is not basic anymore, and is replaced by (i,j).

Example 5.10. We continue our example

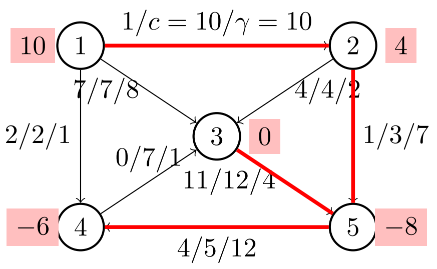


whose BFS is $x_{12} = 1$, $x_{23} = 2$, $x_{35} = 9$, $x_{54} = 4$ with $x_{13} = c_{13} = 7$, $x_{14} = c_{14} = 2$, $x_{25} = c_{25} = 3$ and $x_{43} = 0$.

The edge (2,5) which is currently saturated is activated, it creates the cycle C given by $(5) \leftarrow (2) \rightarrow (3) \rightarrow (5)$. Since (2,5) is saturated, we need to decrease it, which means increasing on the rest of the cycle whose edges are reversed. Then $x_{23} = 2, c_{23} = 4$ and $x_{35} = 9, c_{35} = 12$, which means that on edge (2,3), we could increase the flow by 2, while on the edge (3,5), we could increase the flow by 3. So (2,3) leaves, and (2,5) gets decreased by 2. So $x_{23} = 4$, $x_{25} = 1$ and $x_{35} = 11$.

The sets $\mathcal{B}, \mathcal{S}, \mathcal{N}$ get updated as follows:

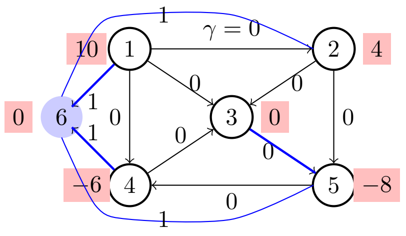
$$\mathcal{B} = \{(1,2),(3,5),(5,4),(2,5)\}, \mathcal{N} = \{(4,3)\}, \mathcal{S} = \{(1,3),(1,4),(2,3)\}.$$



This example has been used to illustrate the different steps of the capacitated simplex network algorithm. See Exercise 47 for a complete solution of this example.

Artificial variables. An initial solution can be used by introducing an artificial node and artificial edges (with as high a capacity as needed), mimicking the uncapacitated case.

Example 5.11. Consider our running example, and suppose we do not know any BFS. We introduce an artificial node, and corresponding artificial edges as shown below. The artificial edges together with the edge $(3,5)$ form a spanning tree. Since it is a capacitated network, we need to assume that the artificial edges have each a capacity which is finite, but we do not specify the actual value of these capacities, e.g., $(1,6)$ has capacity c_{12} and c_{12} is some value higher than whatever flow we will push through $(1,6)$ ($c_{12} \geq 11$ will do here) so the edge is never saturated.



We want to minimize $\gamma^T x$ where γ is a vector with zero everywhere, but for $\gamma_{64} = \gamma_{16} = \gamma_{26} = \gamma_{65} = 1$. We have an immediate BFS given by (note the last equation which describes that the artificial node 6 is a transit node)

$$\begin{array}{ccccc} (1,6) & (3,5) & (6,4) & (2,6) & (6,5) \end{array}$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 \\ -1 & 0 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_{16} \\ x_{35} \\ x_{64} \\ x_{26} \\ x_{65} \end{bmatrix} = \begin{bmatrix} 10 \\ 4 \\ 0 \\ -6 \\ -8 \\ 0 \end{bmatrix}$$

from which we get

$$\begin{bmatrix} x_{16} \\ x_{35} \\ x_{64} \\ x_{26} \\ x_{65} \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ 6 \\ 4 \\ 8 \end{bmatrix}.$$

There is no difference with what we did for the uncapacitated case, we just get the flow to go from the supply nodes to the demand nodes via the artificial edges, the only technical difference is to suppose there are capacities such that we do not saturate edges.

Using complementary slackness, we get:

$$\begin{aligned} u_1 - u_6 &= \gamma_{16} = 1 \\ u_3 - u_5 &\leq \gamma_{35} = 0 \\ u_6 - u_4 &= \gamma_{64} = 1 \\ u_2 - u_6 &= \gamma_{26} = 1 \\ u_6 - u_5 &= \gamma_{65} = 1 \end{aligned}$$

and setting $u_6 = 0$ gives $u_1 = u_2 = 1$, $u_4 = u_5 = -1$ and $u_3 \leq -1$. We then check for feasibility:

$$\begin{aligned} u_1 - u_3 &\geq 2 \not\leq 0 \quad \times \\ u_1 - u_4 &= 2 \not\leq 0 \quad \times \\ u_2 - u_3 &\geq 2 \not\leq 0 \quad \times \\ u_2 - u_5 &= 2 \not\leq 0 \quad \times \end{aligned}$$

so $(1,3)$ is activated, creating the cycle $(1) \rightarrow (3) \rightarrow (5) \leftarrow (6) \leftarrow (1)$, then $x_{13} = 7$ is saturated and:

$$\begin{bmatrix} x_{16} \\ x_{35} \\ x_{64} \\ x_{26} \\ x_{65} \end{bmatrix} = \begin{bmatrix} 3 \\ 7 \\ 6 \\ 4 \\ 1 \end{bmatrix}.$$

This removes the degeneracy in x_{35}, but does not change the edges in the spanning tree. We then activate $(1,4)$, this creates the cycle $(1) \rightarrow (4) \leftarrow (6) \leftarrow (1)$, $x_{14} = 2$ saturates, and

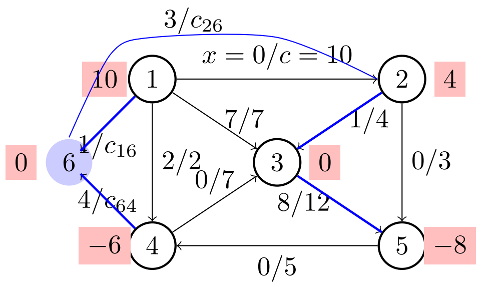
$$\begin{bmatrix} x_{16} \\ x_{35} \\ x_{64} \\ x_{26} \\ x_{65} \end{bmatrix} = \begin{bmatrix} 1 \\ 7 \\ 4 \\ 4 \\ 1 \end{bmatrix}.$$

The spanning tree still has not changed. We activate $(2,3)$, $x_{23} = 1$, this time

$(2,3)$ enters $\mathcal{B}$ and $(6,5)$ leaves:

$$\begin{bmatrix} x_{16} \\ x_{35} \\ x_{64} \\ x_{26} \\ x_{23} \end{bmatrix} = \begin{bmatrix} 1 \\ 8 \\ 4 \\ 3 \\ 1 \end{bmatrix}.$$

At this point, the network flow is as follows:



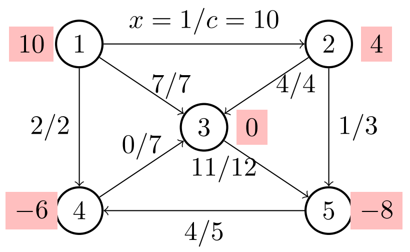
We use complementary slackness once more:

$$\begin{aligned} u_1 - u_6 &= \gamma_{16} = 1 \\ u_3 - u_5 &= \gamma_{35} = 0 \\ u_6 - u_4 &= \gamma_{64} = 1 \\ u_2 - u_6 &= \gamma_{26} = 1 \\ u_2 - u_3 &= \gamma_{23} = 0 \end{aligned}$$

and setting $u_6 = 0$ gives $u_1 = u_2 = u_5 = u_3 = 1$, $u_4 = -1$. We then check for feasibility and $u_5 - u_4 = 2 \not\leq 0$. We activate $(5,4)$, $x_{54} = 3$, then $x_{23} = 4$ saturates, and:

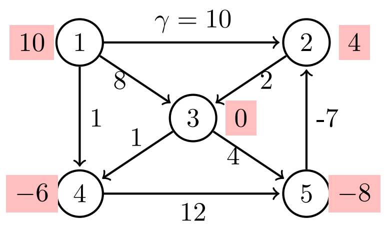
$$\begin{bmatrix} x_{16} \\ x_{35} \\ x_{64} \\ x_{23} \\ x_{54} \end{bmatrix} = \begin{bmatrix} 1 \\ 11 \\ 1 \\ 4 \\ 3 \end{bmatrix}.$$

Thus we are back to a degenerate basis. But now we are left with only 2 artificial edges each with a flow of 1, namely $(1,6)$ and $(6,4)$, so just need to activate $(1,2)$, to get $x_{12} = 1$, $x_{25} = 1$, and :



5.3 Exercises

Exercise 45. Consider the following min cost flow network, where the demands are written at the nodes, and the costs are written next to each edge. Each edge capacity is infinite, and lower bound is 0.

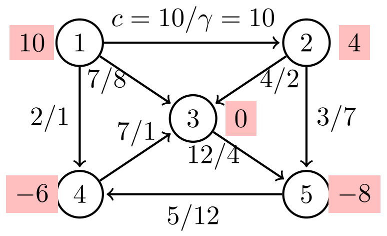


Compute a minimal cost flow for this network, using the BFS $(x_{13}, x_{23}, x_{34}, x_{35}) = (10, 4, 6, 8)$.

Exercise 46. Compute the dual of:

$$\begin{array}{ll} \min & \gamma^T x \\ s.t & \begin{bmatrix} A & 0_m \\ I_m & I_m \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} = \begin{bmatrix} d \\ c \end{bmatrix} \\ & x, s \geq 0. \end{array}$$

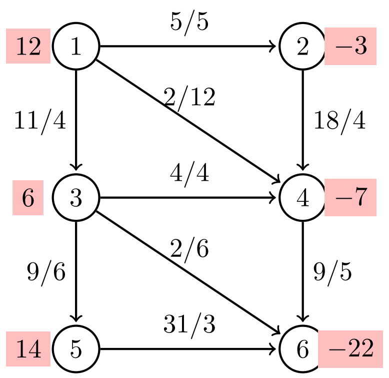
Exercise 47. Consider the following min cost flow problem, where every edge e is given a capacity $c(e)$ and a cost $\gamma(e)$.



1. Compute an optimal solution using the network simplex algorithm.

2. Compute an optimal solution using the negative cycle canceling algorithm.

Exercise 48. Consider the following min cost flow problem, where every edge e is given a capacity $c(e)$ and a cost $\gamma(e)$.



Compute an optimal solution using the network simplex algorithm. Compare the algorithm and the solution with Exercise 33.