Min-cost flow problems and network simplex algorithm *

The particular structure of some LP problems can be sometimes used for the design of solution techniques more efficient than the simplex algorithm. The most relevant case occurs in *min-cost flow problems*. In fact, the particular structure of minimum-cost network flow problems allows for strong simplifications in the simplex method. The following notes assume the reader has basic LP notions, such as the concept of basic feasible solution, the optimality criterion and complementary slackness conditions. The method obtained adapting the simplex method to the structure of flow networks is the *network simplex method*.

1 Min-cost flow problems

The *min-cost flow problem* consists in determining the most economic way to transport a certain amount of good (e.g. oil, oranges, cars ...) from one or more production facilities to one or more consumption facilities, through a given transportation network (e.g. a hydraulic network, a distribution network, a road network etc.). It should be emphasized from the outset that the mathematical model lends itself to represent a variety of problems that have nothing to do with the shipment of goods, and therefore we use the more abstract notion of *flow*.

As usual, the nodes of the network may be associated with physical places (cities, warehouses, industrial facilities, stations ...), and the arcs to one-way communication links (road sections, railways ...) among these places. Note that one does not lose of generality considering the arcs oriented (i.e., one-way) rather than unoriented (i.e., two-way). In fact, each two-way arc can be represented by means of a pair of arcs pointing in opposite directions between the same two nodes. In what follows, given a network $G(N, \mathcal{A})$, we use $n = |N|$ to denote the number of nodes in the network and $m = |\mathcal{A}|$ to denote the number of arcs, so for the graph in Figure 1, $n = 5$ and $m = 8$.

*based on teaching material by Dario Pacciarelli (Dipartimento di Ingegneria, Università di Roma Tre), rearranged by Alessandro Agnetis and Marco Pranzo (Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche, Università di Siena).

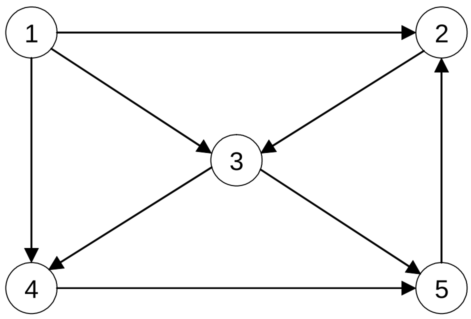

Figure 1: A flow network.

Network topology is only one part of the model. Fundamental information concerns:

- the amount of flow produced or consumed at each node;
- transportation costs between two nodes;
- possibly, an upper limit on maximum flow on each arc (i.e., *capacity*).

We first assume that there are no capacity constraints on arcs, or, in other words, that each arc has infinite capacity. Afterwards (Section 6), we generalize the method to the capacitated case.

For each node i, $i = 1, \ldots, n$, an (integer) number b_i is given, representing the amount of flow produced (if $b_i > 0$) or consumed (if $b_i < 0$) at i. The nodes that *produce* flow are sometimes referred to as *sources*, and b_i as *supply*. Nodes that *consume* flow are called *sinks*, and $|b_i|$ as *demand*. If $b_i = 0$, node i does not consume nor produce flow, i.e., it is a *transit node*. Note that this classification into three types of nodes is completely independent of the structure of the network, but it is defined only by the values of supply and demand. For the example in Figure 1, assume there is a demand of 6 units at node 4 and of 8 units at node 5, while there is a supply of 10 units at node 1, and 4 units at node 2, i.e.:

$$b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{pmatrix} = \begin{pmatrix} 10 \\ 4 \\ 0 \\ -6 \\ -8 \end{pmatrix} \tag{1}$$

Notice that nodes 4 and 5 are sinks, nodes 1 and 2 are sources and node 3 is a transit node.

We assume that the following assumption is always valid:

Assumption 1.1 *Total supply equals total demand.*

At very first glance, this assumption appears rather unrealistic in practice: it would actually be surprising that the total demand for oranges perfectly matches the supply to the market. The key issue is that this assumption does not limit the applicability of the theoretical results. In fact, Assumption 1.1 can always be met by adding appropriate nodes and arcs, having the same role of slack and surplus variables used to write LP problems in standard form. On the other hand, Assumption 1.1 simplifies theoretical developments.

As for the transportation cost from one location to another, it is assumed that each arc (i, j) of the network has an associated *unit transport cost* c_{ij}. Therefore, for each unit of flow sent from a source to a destination, a cost is incurred equal to the sum of unit transport costs of all the arcs traversed. The *min-cost flow problem* consists in finding a solution that minimizes the total cost while meeting the demand of all nodes in the network. For the example of Figure 1, a possible cost vector is the following:

$$c = \begin{pmatrix} c_{12} \\ c_{13} \\ c_{14} \\ c_{23} \\ c_{34} \\ c_{35} \\ c_{45} \\ c_{52} \end{pmatrix} = \begin{pmatrix} 10 \\ 8 \\ 1 \\ 2 \\ 1 \\ 4 \\ 12 \\ -7 \end{pmatrix} \tag{2}$$

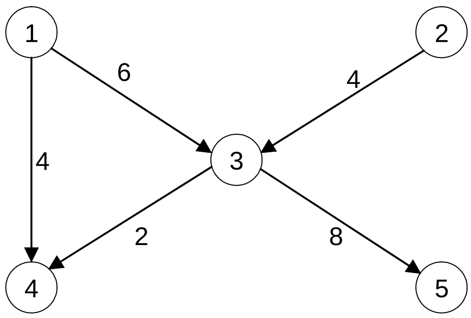

Figure 2: A feasible flow.

In a min-cost flow problem, a solution is defined by specifying the *flow* x_{ij} in each arc (i, j) of the network. A solution can then be represented by a vector x having m components. A solution is *feasible* if and only if:

(i) for each transit node, total incoming flow equals total outgoing flow.

(ii) for each sink, the total incoming flow equals node demand plus total outgoing flow.

(iii) For each source, the total outgoing flow equals node supply plus total incoming flow.

(iv) All arc flows are non-negative.

Conditions (i)–(iii) are encompassed by the following constraints, for each node i:

$$\sum_{(i,j)\in\delta^+(i)} x_{ij} - \sum_{(j,i)\in\delta^-(i)} x_{ji} = b_i$$

For example, one can easily check that for the problem in Figure 1 and the vector b given in (1), a feasible flow is:

$$x = \begin{pmatrix} x_{12} \\ x_{13} \\ x_{14} \\ x_{23} \\ x_{25} \\ x_{34} \\ x_{35} \\ x_{45} \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \\ 4 \\ 4 \\ 0 \\ 2 \\ 8 \\ 0 \end{pmatrix} \tag{3}$$

The problem consists in determining a flow vector x that minimizes cost $c^T x$ among all feasible solutions. These conditions, together with the objective function of the problem, can be expressed as:

$$\begin{aligned} \min \quad & c^T x \\ & Ax = b \\ & x \geq 0 \end{aligned} \tag{4}$$

where A is the incidence matrix of the network. Matrix A has size $n \times m$, hence (4) is an LP with m variables and n constraints. For the example in Figure 1, matrix A and the min-cost flow LP are:

$$A = \begin{bmatrix} 1 & 1 & 1 & & & & & \\ -1 & & & 1 & & & & -1 \\ & -1 & & -1 & 1 & 1 & & \\ & & -1 & & -1 & & 1 & \\ & & & & & -1 & -1 & 1 \end{bmatrix} \tag{5}$$

$$\begin{array}{llcccccccccl} \min & \multicolumn{11}{l}{10x_{12} + 8x_{13} + x_{14} + 2x_{23} + x_{34} + 4x_{35} + 12x_{45} - 7x_{52}} \\ & x_{12} & +x_{13} & +x_{14} & & & & & & = & 10 \\ & -x_{12} & & & +x_{23} & & & & -x_{52} & = & 4 \\ & & -x_{13} & & -x_{23} & +x_{34} & +x_{35} & & & = & 0 \\ & & & -x_{14} & & -x_{34} & & +x_{45} & & = & -6 \\ & & & & & & -x_{35} & -x_{45} & +x_{52} & = & -8 \\ & x \geq 0 \end{array} \tag{6}$$

Note that A is *not* full-rank: adding all rows one obtains the null vector, i.e., the rows of A are not linearly independent. Assumption 1.1 ensures that the problem *does* have solution, since $rank[A] = rank[A \;\; b]$. Hence, in a min-cost flow problem, one of the equations can be canceled.

For the sake of simplicity, when representing a feasible solution, we often depict only the arcs (i, j) for which $x_{ij} > 0$, omitting the arcs (i, j) such that $x_{ij} = 0$. The flow specified by (3) can therefore be represented as in Figure 2.

In the following sections 2–5we show how the simplex method can be specialized to solve uncapacitated min-cost flow problems. In Section 6 we generalize it to the capacitated case.

2 Basic solutions and spanning trees

Since we know that the matrix A is not full-rank, a basis of A consists of only $n - 1$ linearly independent columns of A. These columns correspond to a collection of arcs of the flow network. We want to show that, if the network is connected (as we will always suppose), all the bases of A are associated with *spanning trees*. By spanning tree here we mean a set of $n - 1$ arcs such that each node of the network is adjacent to at least one of them (it coincides with the classical notion of spanning tree if one disregards arc orientation). Also, throughout these notes we always use the term *cycle* to refer to a set of arcs forming a closed path (i.e., a path in which the first and the last node of the path coincide) *when ignoring their orientation*. Figure 3 shows an example of a spanning tree for the graph of Figure 1.

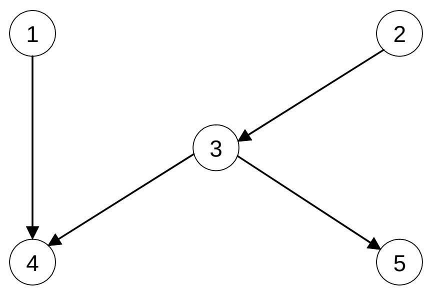

Figure 3: A spanning tree.

First of all let us prove the following lemma:

Lemma 1 *Let $T \subset A$ be a set of columns of A such that the corresponding arcs form a spanning tree. Then, the columns of T are linearly independent.*

Proof – We begin by observing that every tree has the following properties: regardless of how an initial node v_0 is labeled, the remaining nodes can be numbered $v_1, v_2, .., v_{n-1}$ in such a way that, for each $i \geq 1$, there is exactly one arc with one endpoint equal to v_i and the other endpoint equal to one of the previously labeled nodes $v_0, v_1, .., v_{i-1}$. To this aim, it is sufficient to visit the tree and label the nodes (from v_0 to v_{n-1}) and the arcs (from 1 to $n-1$) in the order in which they are visited. If we order the $n-1$ columns of the spanning tree and the n rows of A following the above ordering, we obtain a matrix $n \times (n-1)$. If we discard the first row of such matrix (associated with the label v_0), we are left with a submatrix of A of size $(n-1) \times (n-1)$ which, by construction, is upper triangular and has all nonzero elements on the main diagonal. Hence, the determinant of this submatrix is different from zero, which implies that the matrix A has rank $n-1$, and that the columns associated with a spanning tree are always linearly independent. This demonstrates that the columns of A associated with a spanning tree of the network of flow are always bases of A. $\square$

As an example, for the spanning tree of Figure 3, visiting the nodes in the order $1, 4, 3, 2, 5$, and hence the arcs in the order $(1,4), (3,4), (2,3), (3,5)$, we obtain the new numbering of Figure 4. The corresponding incidence matrix, re-ordering rows and columns, is:

$$
\begin{array}{c} \textit{label} \\ \\ v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array}
\begin{array}{c} \textit{arcs} \\ (1,4)(3,4)(2,3)(3,5) \\ \left[\begin{array}{cccc} 1 & & & \\ -1 & -1 & & \\ & 1 & -1 & 1 \\ & & 1 & \\ & & & -1 \end{array}\right] \end{array}
\qquad (7)
$$

In the remainder of this section, we want to show that the viceversa of Lemma 1 also holds:

Lemma 2 *If a subset T of columns of A is a basis, then the corresponding arc set is a spanning tree.*

Proof – A basis of A necessarily contains $n-1$ columns, and an arc set consisting of $n-1$ arcs that does not correspond to a spanning tree must necessarily contain a cycle. For our purposes, it is therefore sufficient to show that the columns of A associated with a cycle are linearly dependent, i.e., that there is a linear combination of the columns, with

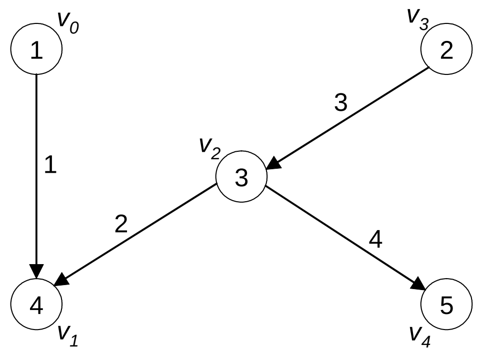

Figure 4: A labeled tree.

coefficients not all zero, yielding the null vector. Given a cycle, the coefficients of the linear combination can be simply obtained as follows. Arbitrarily choose one direction for the cycle, and set the coefficient of an arc (i, j) of the cycle as:

- 1 if the arc has the *same* orientation as the direction chosen;
- -1 if the arc has the *opposite* orientation as the direction chosen;

It is easy to verify that this linear combination of the columns of A associated yields the null vector. □

As an example, consider the cycle formed by arcs $(1, 3)$, $(3, 4)$ and $(1, 4)$ in Figure 2. Visiting the cycle in the verse $1, 3, 4$, one has that $(1, 3)$ and $(3, 4)$ are oriented like the visiting verse, while $(1, 4)$ has opposite orientation. The columns of A associated with the three arcs are:

$$A_{13} = \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} \quad A_{34} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \\ 0 \end{bmatrix} \quad A_{14} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

Adding A_{13} to A_{34} and subtracting A_{14}, one gets the null vector, hence showing that these three columns are linearly dependent, hence not all of them can appear in the same basis of A.

In conclusion, Lemmas 1 and 2 imply the following fundamental result:

THEOREM 1 *Given a connected flow network, letting A be its incidence matrix, a submatrix B of size $(n-1) \times (n-1)$ is a basis of A if and only if the arcs associated with the columns of B form a spanning tree.*

Solving a min-cost flow problem with the simplex algorithm, one has therefore that all the basic feasible solutions explored by the algorithm are spanning trees of the flow network. As it occurs for any LP, also in min-cost flow problems one has feasible, infeasible and degenerate bases. A basis is feasible if $x_B = B^{-1}b \geq 0$. In this case, it can be easily verified solving the system $Bx_B = b$, starting from a leaf of the spanning tree, and verifying that $x_B \geq 0$. For instance, one can easily verify that the tree in Figure 3 is *not* feasible, since the corresponding basic solution is:

$$x_B = \begin{pmatrix} x_{14} \\ x_{34} \\ x_{23} \\ x_{35} \end{pmatrix} = \begin{pmatrix} 10 \\ -4 \\ 4 \\ 8 \end{pmatrix}$$

A basic feasible solution is shown in Figure 5, for which one has:

$$x_B = \begin{pmatrix} x_{13} \\ x_{23} \\ x_{34} \\ x_{35} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ -b_4 \\ -b_5 \end{pmatrix} = \begin{pmatrix} 10 \\ 4 \\ 6 \\ 8 \end{pmatrix} \tag{8}$$

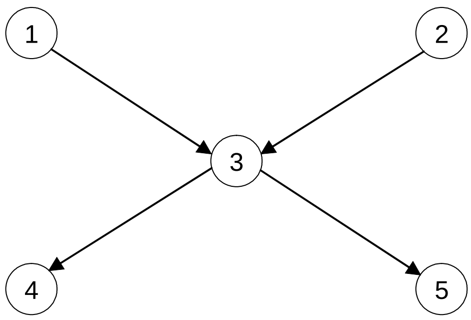

Figure 5: A basic feasible solution.

3 Optimality criterion

In this section we show how to efficiently check the optimality of a basic feasible solution, using complementarity slackness conditions. The dual of the min-cost flow problem (4) is the following:

$$\begin{aligned} \max \quad & b^T u \\ & u^T A \leq c^T \end{aligned} \tag{9}$$

Actually, as already discussed, the primal problem has a redundant equation, which could be eliminated without altering the problem. This means that the dual has one redundant variable, which could be eliminated without altering the problem. For notation simplicity, we will not explicitly eliminate equations or variables from the primal and dual problems, but we will account for this observation by arbitrarily setting one of the dual variables to zero (which is equivalent to deleting it). The dual problem can be conveniently written in full as:

$$\begin{array}{ll} \max & \sum_{i=1}^{n} b_i u_i \\ & u_i - u_j \leq c_{ij} \qquad \forall\,(i,j) \in \mathcal{A} \end{array} \tag{10}$$

The dual variables u are often called *potentials*. Consider then a basic feasible solution, and partition $\mathcal{A}$ into two sets $\mathcal{B}$ (the arcs of the spanning tree) and $\mathcal{F}$ (the other arcs of the network). If an arc is basic in the optimal solution, the corresponding dual constraint must be satisfied at equality, i.e.:

$$u_i - u_j = c_{ij} \qquad \forall\,(i,j) \in \mathcal{B} \tag{11}$$

Equations (11) are $n-1$, in n variables. As previously discussed, one variable can be arbitrarily fixed to zero. One can then easily determine a solution u, and check its dual feasibility, i.e., the fulfilment of all other constraints:

$$u_i - u_j \leq c_{ij} \qquad \forall\,(i,j) \in \mathcal{F} \tag{12}$$

Hence if, given a basic feasible solution x, the values u_i obtained through (11) also respect the (12), x is optimal.

For instance, for the feasible basis in Figure 5, and with the costs in (6), one has:

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

From this, arbitrarily letting $u_3 = 0$, one has $u_1 = 8$, $u_2 = -2$, $u_4 = -1$, $u_5 = -4$. Plugging these values into (12), one gets:

$$\begin{cases} u_1 - u_2 = 10 \leq c_{12} = 10 \\ u_1 - u_4 = 9 \not\leq c_{14} = 1 \\ u_4 - u_5 = 3 \leq c_{45} = 12 \\ u_5 - u_2 = -2 \not\leq c_{52} = -7 \end{cases}$$

Hence, the first and third constraint are satisfied, while the second and the fourth are violated. Therefore, the corresponding basis is *not* optimal.

4 Unboundedness and pivot operation

We now want to show how it is possible to either profitably change the basis or verify that the problem is unbounded.

If a solution x does not verify the optimality conditions, by (12) there must exist an arc $(i, j) \in \mathcal{F}$ such that $u_i - u_j > c_{ij}$. In other words, the reduced cost of the variable x_{ij}, letting A_{ij} be the column associated with that variable, is

$$\bar{c}_{ij} = c_{ij} - u^T A_{ij} = c_{ij} - u_i + u_j < 0,$$

and then it will be profitable to bring variable x_{ij} into the basis, i.e., *activate* arc (i, j). Clearly, arc (i, j) forms a cycle with the arcs of $\mathcal{B}$ (the spanning tree). In order to restore a feasible basis, we must get rid of this cycle, i.e., letting $\mathcal{C}$ denote the arcs of the cycle, the arc that *leaves* the basis must necessarily be an arc of $\mathcal{C}$. Which arc must then leave the basis? We can proceed in a completely analogous manner to what is done for the general simplex method. In fact, since $\bar{c}_{ij} < 0$, it is profitable to increase the flow on (i, j). This implies that, in order to maintain the feasibility of the current solution, one must necessarily alter the value of the flow on all the arcs of $\mathcal{C}$, increasing the flow on arcs that, in the cycle, have the same orientation as (i, j) (*direct arcs*), and decreasing the flow of the arcs having opposite orientation (*reverse arcs*).

A special situation occurs if *all* the arcs in $\mathcal{C}$ are direct. In this case, one can indefinitely increase the flow on the arcs of $\mathcal{C}$, while simultaneously decreasing the objective function. In this case the problem is *unbounded.* This condition occurs if and only if in the *directed* cycle $\mathcal{C}$ the sum of the costs of the arcs is negative:

$$\sum_{(h,k)\in\mathcal{C}} c_{hk} < 0.$$

If we are not in this situation, there is at least one reverse arc. As the flow on (i, j) is increased, the flow in reverse arcs decreases, and there is at least one arc, say (u, v), such that its flow reaches zero before all the other arcs, i.e., arc (u, v) leaves the basis. The maximum feasible value ϑ of the flow in arc (i, j) is therefore:

$$\vartheta = \min\{x_{hk} : (h, k) \in \mathcal{C},\ \ (h, k)\ \ is\ \ a\ \ reverse\ \ arc\}.$$

The new basic feasible solution is therefore obtained by simply *increasing* by ϑ the flow in direct arcs and *decreasing* by the same amount ϑ the flow in reverse arcs of $\mathcal{C}$.

Still continuing with the Example of Figure 1 and with the basis of Figure 5, a possible arc entering the basis is $(i, j) = (1, 4)$, which forms cycle $\mathcal{C} = \{(1, 4), (1, 3), (3, 4)\}$. Arcs $(1, 3)$ and $(3, 4)$ are reverse arcs $(1, 4)$. The flow in these arcs, given by (8), is $x_{13} = 10$ and

$x_{34} = 6$. Therefore $\vartheta = 6$, the arc leaving the basis is $(3,4)$ and the new basic variables are:

$$\tilde{x}_B = \begin{pmatrix} \tilde{x}_{13} \\ \tilde{x}_{23} \\ \tilde{x}_{14} \\ \tilde{x}_{35} \end{pmatrix} = \begin{pmatrix} 10-6 \\ 4 \\ 0+6 \\ 8 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \\ 6 \\ 8 \end{pmatrix} \tag{13}$$

The pivot operation is highlighted in Figure 6. We can now proceed with the new iteration:

Computation of variables u. From complementary slackness conditions one has:

$$\begin{cases} u_1 - u_3 = c_{13} = 8 \\ u_2 - u_3 = c_{23} = 2 \\ u_1 - u_4 = c_{14} = 1 \\ u_3 - u_5 = c_{35} = 4 \end{cases}$$

Arbitrarily fixing $u_1 = 0$ one has $u_3 = -8$, $u_2 = 2 - 8 = -6$, $u_4 = -1$, $u_5 = -12$. Dual feasibility requires:

$$\begin{cases} u_1 - u_2 \leq c_{12} = 10 \\ u_3 - u_4 \leq c_{34} = 1 \\ u_4 - u_5 \leq c_{45} = 12 \\ u_5 - u_2 \leq c_{52} = -7 \end{cases}$$

Note that the fourth condition is violated. Hence, arc $(5,2)$ enters the basis and creates the cycle $\mathcal{C} = \{(5,2),(2,3),(3,5)\}$. Since all the arcs of $\mathcal{C}$ have the same orientation as $(5,2)$, we conclude that the problem is unbounded, and there is no optimal solution. In fact, the cycle $\{(5,2),(2,3),(3,5)\}$ has total cost:

$$c_{52} + c_{23} + c_{35} = 2 + 4 - 7 = -1 < 0$$

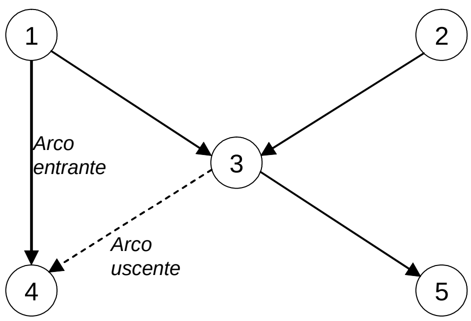

Figure 6: Pivot operation.

5 Phase 1 of the network simplex method

We have so far neglected the problem of determining a feasible flow to start with. This problem is taken care by the *phase 1* of the network simplex method.

The idea is quite simple. We recall that in phase 1 of the simplex method an artificial variable is added for each constraint, so that a feasible basis for the artificial problem is always obtained. In the network simplex method, one can operate in a perfectly analogous way. This time, variables are associated with the arcs of a flow network, so we have to add *artificial arcs* in order to immediately get a feasible solution. The simplest way to obtain this is to convey all the flow produced by sources into an artificial node via artifical arcs, and redistribute such flow via other artificial arcs connecting the new node to sinks. Note that, if there is at least one transit node, artificial arcs are not enough to form a spanning tree (of the artificial network). To obtain an initial basis one can then add an arbitrary arc of the original network for each transit node, so that no loops are formed among them and with the artificial arcs.

Similar to the general simplex method, also in the phase 1 of the network simplex method the objective is to minimize the flow on artificial arcs, so in the artificial problem we set the cost of each artificial arc equal to 1 and the cost of all other arcs to 0.

Once the optimal solution to the artificial problem has been determined, a feasible flow (if it exists) for the original problem can be easily obtained. Notice that if, in the optimal solution of the artificial problem, some artificial arc has nonzero flow, the original problem is *infeasible*. If the flow in all the artificial arcs is zero, these can be simply removed (along with the artificial node), since the flow in the original arcs is a feasible flow for the original problem. Possibly, some (empty) arc may need to be added to the basis to restore a spanning tree, and start phase 2. In the latter case, the initial basis is degenerate.

For example, consider the network in Figure 7 and the demand vector (14).

$$d = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{pmatrix} = \begin{pmatrix} 2 \\ -6 \\ 0 \\ 8 \\ -4 \end{pmatrix} \tag{14}$$

Phase 1 is set up adding an artificial node 6, two artificial arcs $(1, 6)$ and $(4, 6)$ from the two sources to the artificial node, and two artificial arcs $(6, 2)$ and $(6, 5)$ from the artificial node to the two sinks, as in Figure 8. To obtain an initial basis for the problem, it is necessary to add another arc to connect the transit node 3 to the rest of the tree, e.g., arc $(3, 5)$. The objective function is to minimize the sum of the flows on the artificial

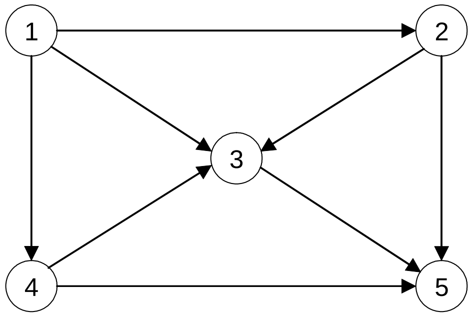

Figure 7: Example of transportation network.

arcs. The cost vector is given by (2), along with the initial basic solution. The initial spanning tree is reported in Figure 9.

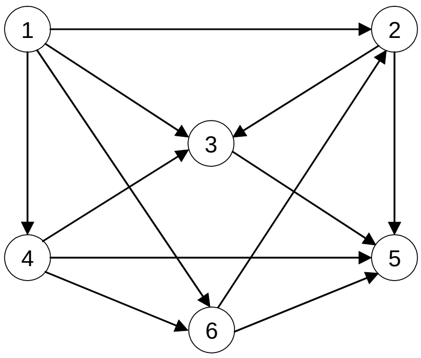

Figure 8: Artificial network.

$$c = \begin{pmatrix} c_{12} \\ c_{13} \\ c_{14} \\ c_{23} \\ c_{25} \\ c_{35} \\ c_{43} \\ c_{54} \\ c_{16} \\ c_{46} \\ c_{62} \\ c_{65} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \qquad x_B = \begin{pmatrix} x_{16} \\ x_{46} \\ x_{62} \\ x_{65} \\ x_{35} \end{pmatrix} = \begin{pmatrix} |b_1| \\ |b_4| \\ |b_6| \\ |b_5| \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 8 \\ 6 \\ 4 \\ 0 \end{pmatrix} \tag{15}$$

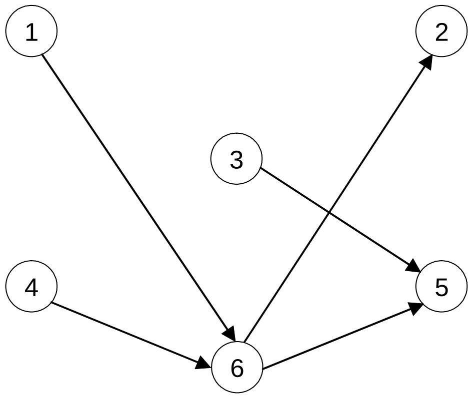

Figure 9: A feasible basis for the artificial problem.

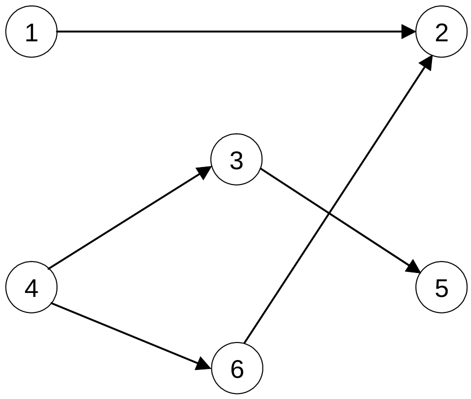

Figure 10: An optimal basis for the artificial problem which is infeasible for the original problem.

The reader can verify that, if arc $(1, 2)$ enters the basis, arc $(1, 6)$ leaves. Thereafter, if we let $(4, 3)$ enter the basis, $(6, 5)$ leaves. At this point, the solution is optimal for phase 1. However, the artificial arcs $(4, 6)$ and $(6, 2)$ remain in basis, having flow $x_{46} = x_{62} = 4$. Hence, the original problem is impossible. The optimal spanning tree for phase 1 is shown in Figure 10.

This situation can also be directly interpreted on the network. In fact, if we consider a dual solution such that $u_6 = 1$, the other nodes of the network are divided into nodes with dual variable value $u_i = 2$ and nodes with dual variable value $u_j = 0$. If the solution is optimal for phase 1, there cannot exist an arc (i, j) in the network that goes from one node with dual value 2 to a node with dual value 0, since this would violate the dual feasibility $u_i - u_j \leq 0$). However, the sum of the demands of the nodes with dual value 0 is strictly negative, while the sum of the demands of the nodes with dual value 2 is strictly positive. In other words, there is an overall flow produced by nodes with dual value 0, which can not reach the nodes with dual value 2, and this determines the impossibility to solve the problem. In the example of Figure 7 one can see that node 2 (sink) is unreachable from node 4 (source), and node 1 is unable to satisfy all the demand of node 2.

6 Capacitated networks

In this section we illustrate how to extend the network simplex algorithm to capacitated networks, i.e., when there are capacities associated with arcs on the network. The case of capacitated networks is fairly frequent in the applications, since there is typically a limit to the amount of flow in an arc.

Let k_{ij} denote the capacity of arc (i, j), i.e., in a feasible flow vector, $0 \leq x_{ij} \leq k_{ij}$. We let k denote the capacity vector. The min-cost flow problem on a capacitated network can therefore be formulated as follows:

$$\begin{array}{l} \min c^T x \\ Ax = b \\ x \leq k \\ x \geq 0 \end{array} \tag{16}$$

adding a vector s of slack variables, the problem can be put into standard form:

$$\begin{array}{l} \min[\ c \quad 0\ ] \begin{bmatrix} x \\ s \end{bmatrix} \\ \begin{bmatrix} A & 0 \\ I & I \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} = \begin{bmatrix} b \\ k \end{bmatrix} \\ x, s \geq 0 \end{array} \tag{17}$$

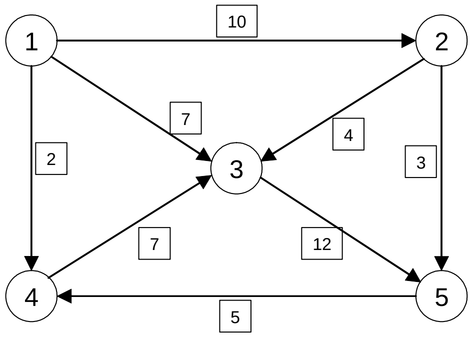

Figure 11: Capacitated flow network.

Figure 11 shows the network flow of Figure 7, with the addition of arc capacities (indicated in rectangles). Problem data are:

$$k = \begin{pmatrix} k_{12} \\ k_{13} \\ k_{14} \\ k_{23} \\ k_{25} \\ k_{35} \\ k_{43} \\ k_{54} \end{pmatrix} = \begin{pmatrix} 10 \\ 7 \\ 2 \\ 4 \\ 3 \\ 12 \\ 7 \\ 5 \end{pmatrix} \qquad c = \begin{pmatrix} c_{12} \\ c_{13} \\ c_{14} \\ c_{23} \\ c_{25} \\ c_{35} \\ c_{43} \\ c_{54} \end{pmatrix} = \begin{pmatrix} 10 \\ 8 \\ 1 \\ 2 \\ 7 \\ 4 \\ 1 \\ 12 \end{pmatrix} \qquad b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{pmatrix} = \begin{pmatrix} 10 \\ 4 \\ 0 \\ -6 \\ -8 \end{pmatrix} \tag{18}$$

Let us now review all steps of the network simplex algorithm to extend them to the capacitated case.

In the uncapacitated case, basic solutions correspond to spanning trees, and can be used to partition the set of arcs $\mathcal{A}$ into the two subsets $\mathcal{B}$ (basic arcs) and $\mathcal{F}$ (nonbasic arcs). In the capacitated case, the correspondence between trees and spanning trees and basic solutions must be slightly revised. In particular, given a basic feasible solution x to the problem, we can partition the arc set into *three* subsets: the set $\mathcal{V}$ of empty arcs (i.e., arcs with zero flow), the set $\mathcal{S}$ of saturated arcs (i.e., with flow $x_{ij} = k_{ij}$), and the set, call it $\mathcal{B}$, of arcs that are neither saturated nor empty.

Theorem 2 *Given a connected flow network, a basic feasible solution x and the corresponding tripartition $(\mathcal{V}, \mathcal{S}, \mathcal{B})$ of the arcs, the arcs of $\mathcal{B}$ do not form cycles.*

Proof – Suppose that the arcs of $\mathcal{B}$ form a cycle. Only for ease of exposition, let us refer to a cycle consisting of five arcs (of course, the conclusions will hold for cycles with any number of arcs):

$(i_1, i_2), (i_2, i_3), (i_4, i_3), (i_4, i_5), (i_1, i_5)$

in which, arbitrarily fixing as verse of the cycle that of arc (i_1, i_2), some arcs (namely $(i_1, i_2), (i_2, i_3), (i_4, i_5)$) are consistent with the verse of the cycle (*direct* arcs), while the others (*reverse* arcs, such as $(i_4, i_3), (i_1, i_5)$) are not.

Referring to formulation (17), variables $\{x_{i_1,i_2}, x_{i_2,i_3}, x_{i_4,i_5}\}$ are certainly basic, and hence so are slack variables $\{s_{i_1,i_2}, s_{i_2,i_3}, s_{i_4,i_5}\}$, since the arcs in $\mathcal{B}$ are not saturated. The 10 columns corresponding to such variables (reporting only the rows of these columns with nonzero elements) are:

$$
\begin{array}{c}
\\
i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ (i_1,i_2) \\ (i_2,i_3) \\ (i_4,i_3) \\ (i_4,i_5) \\ (i_1,i_5)
\end{array}
\left[
\begin{array}{cccccccccc}
x_{i_1,i_2} & x_{i_2,i_3} & x_{i_4,i_3} & x_{i_4,i_5} & x_{i_1,i_5} & s_{i_1,i_2} & s_{i_2,i_3} & s_{i_4,i_3} & s_{i_4,i_5} & s_{i_1,i_5} \\
1 & & & & 1 & & & & & \\
-1 & 1 & & & & & & & & \\
 & -1 & -1 & & & & & & & \\
 & & 1 & 1 & & & & & & \\
 & & & -1 & -1 & & & & & \\
\hline
1 & & & & & 1 & & & & \\
 & 1 & & & & & 1 & & & \\
 & & 1 & & & & & 1 & & \\
 & & & 1 & & & & & 1 & \\
 & & & & 1 & & & & & 1
\end{array}
\right]
\tag{19}
$$

Now perform a linear combination of the columns, using the following coefficients. In the combination, use coefficient +1 for the columns of variables x_{ij} corresponding to direct arcs and of variables s_{ij} corrisponding to reverse arcs, and use coefficient -1 for the columns of variables x_{ij} corrisponding to reverse arcs and of variables s_{ij} corresponding to direct arcs. In this way, we obtain the null column. This shows that the 10 columns indicated cannot be all in the basis, and hence the 5 corresponding arcs cannot be all in $\mathcal{B}$. $\square$

From the above result, it follows that, in a basic solution, at most $n-1$ arcs can belong to $\mathcal{B}$, and these arcs cannot form cycles. This fact allows us to continue identifying basic solutions with spanning trees. However, note that this time more basic solutions may correspond to the same spanning tree (set $\mathcal{B}$). In fact, we get different basic solutions for different sets of saturated and empty arcs ($\mathcal{V}$ and $\mathcal{S}$). It is quite simple in fact to verify that, given the sets $\mathcal{V}$ and $\mathcal{S}$, the value of the flow in the other arcs is uniquely determined and can be computed through the same procedure already described for the uncapacitated case. One should only observe that, if $(i, j) \in \mathcal{S}$, it must be taken into account when calculating the flow on the arcs of $\mathcal{B}$. For example, consider the network in Figure 11, and let $\mathcal{B} = \{(1, 2), (2, 3), (3, 5), (5, 4)\}$, while $\mathcal{S} = \{(1, 3), (1, 4), (2, 5)\}$ and $\mathcal{V} = \{(4, 3)\}$ (Figure 12). Clearly, $x_{43} = 0$, $x_{13} = 7$, $x_{14} = 1$ and $x_{25} = 3$. Subsequently one can compute $x_{12} = 10 - 2 - 7 = 1$, $x_{23} = 4 + 1 - 3 = 2$, $x_{35} = 8 + 4 - 3 = 9$ and $x_{54} = 9 + 3 - 8 = 4$. Note that all these values satisfy capacity constraints, hence the solution is feasible.

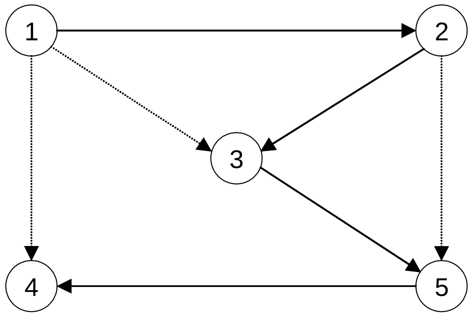

Figure 12: Sets $\mathcal{B}$ (solid line) and $\mathcal{S}$ (dashed line).

Moreover, note that – analogously to the uncapacitated case – also in $\mathcal{B}$ there can be saturated or empty arcs. In this case, one has a *degenerate basic solution* (and a corresponding *degenerate spanning tree*). As in the uncapacitated case, saturated or empty arcs added to complete the basis, are considered exactly as all the arcs belonging to $\mathcal{B}$.

Let us now extend the optimality criterion to the capacitated case. First, write the dual of (17) as:

$$\begin{cases} \max[\ d^T \quad k^T\ ] \begin{bmatrix} u \\ v \end{bmatrix} \\ \begin{bmatrix} u^T\ v^T \end{bmatrix} \begin{bmatrix} A & 0 \\ I & I \end{bmatrix} \leq [c^T\ 0^T] \end{cases} \tag{20}$$

Completarity slackness conditions are:

$$\left([c^T\ 0^T] - [u^T\ v^T] \begin{bmatrix} A & 0 \\ I & I \end{bmatrix} \right) \begin{bmatrix} x \\ s \end{bmatrix} = 0 \tag{21}$$

For each arc (i, j), one has therefore the two conditions:

$$(c_{ij} - u_i + u_j - v_{ij})\ x_{ij} = 0 \tag{22}$$

$$v_{ij}\ s_{ij} = 0 \tag{23}$$

If $(i, j) \in \mathcal{B}$, then $x_{ij} \neq 0$ and $s_{ij} \neq 0$, and so (22) and (23) reduce to

$$c_{ij} - u_i + u_j = 0 \quad (i, j) \in \mathcal{B} \tag{24}$$

analogously to (12). These are then $n-1$ equations in n variables, one of which, as usual, can be arbitrarily fixed to 0. The values found must satisfy all dual constraints. Hence, considering an empty arc $(i,j) \in \mathcal{V}$, since $s_{ij} \neq 0$, one has (from (23)) $v_{ij} = 0$ and the following must hold

$$u_i - u_j \leq c_{ij} \quad (i,j) \in \mathcal{V} \tag{25}$$

For a saturated arc, from (22) one has:

$$c_{ij} - u_i + u_j = v_{ij}$$

and therefore, considering that, from (20) one has $v_{ij} \leq 0$, it must hold

$$u_i - u_j \geq c_{ij} \quad (i,j) \in \mathcal{S} \tag{26}$$

Summarizing, at each iteration of the network simplex algorithm, the current basis is optimal if the potentials at nodes u_i, computed using (24), satisfy all inequalities (25) and (26). For the example in Figure 12, the potentials have the following values, obtained by fixing the potential of node 4 to zero ($u_4 = 0$):

$$u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix} = \begin{pmatrix} 28 \\ 18 \\ 16 \\ 0 \\ 12 \end{pmatrix} \tag{27}$$

In this example, the solution is not optimal, since not all arcs in $\mathcal{S} \cup \mathcal{V}$ satisfy the optimality conditions. In particular, for the saturated arc $(2,5)$ one has $\bar{c}_{25} = u_2 - u_5 - c_{25} = 18 - 12 - 7 = -1 \ngeq 0$.

If the optimality conditions are not met, any arc (empty or full) that does not meet its optimality condition may be chosen to enter the basis. Once the arc entering the basis is selected, one must determine the arc leaving the basis. Generalizing what done in the uncapacitated case, one must find the maximum flow ϑ circulating in the cycle $\mathcal{C}$ generated by the addition of the arc (i,j). In this case, the maximum circulating flow ϑ may be limited by the capacity of direct arcs which reach saturation, as well as by reverse arcs that are emptied.

$$\vartheta = \min \left\{ \begin{array}{c} x_{pq} : (p,q) \in \mathcal{C}, (p,q) \ \ is \ \ a \ \ direct \ arc, \\ k_{pq} - x_{pq} : (p,q) \in \mathcal{C}, (p,q) \ \ is \ \ a \ \ reverse \ arc \end{array} \right\} \tag{28}$$

The arc that determines the value ϑ leaves the basis, and is replaced with (i,j). In particular, tree $\mathcal{B}$ is updated exactly as in the uncapacitated case, i.e., by suitably increasing or decreasing by ϑ the flow on the arcs of $\mathcal{C}$. (Note that, unlike the capacitated case, if all capacities are finite, it is not possible to have unbounded solutions.) An iteration of the simplex method is called *not degenerate* if the flow $\vartheta > 0$, while an iteration is called *degenerate* if $\vartheta = 0$. Degenerate iterations can happen only if the spanning tree $\mathcal{B}$ is degenerate, i.e., if it contains saturated or empty arcs. With reference to the network and the basic solution in Figure 12, when arc $(1,4)$ enters the basis, the cycle is $\mathcal{C} = \{(2,3),(3,5),(2,5)\}$. As the flow on $(2,5)$ decreases, the first arc to leave $\mathcal{B}$ is $(2,3)$, which becomes saturated, and $\vartheta = 2$. As a consequence, in the new basic feasible solution, $x_{25} = 2, x_{23} = 4$ and $x_{35} = 11$. Hence, the new partition of the arcs is: $\mathcal{B}' = \{(1,2),(2,5),(3,5),(5,4)$, $\mathcal{S}' = \{(1,3),(1,4),(2,3)\}$ and $\mathcal{V}' = \{(4,3)\}$. We leave it as an exercise to verify that the new solution is optimal.

Finally, note that for the computation of an initial feasible solution (phase 1) it is possible to proceed exactly as with uncapacitated networks (specifying a sufficiently large capacity for artificial arcs).