Lecture Notes for the course "Design and Operation of Traffic and Telecommunication Networks"

(Version 0.1, Lectures of January 8th - 15th 2016)

Bachelor of Science in Mathematics
Freie Universität Berlin

by
Fabio D'Andreagiovanni
Department of Mathematical Optimization, Zuse Institute Berlin (ZIB)
d.andreagiovanni@zib.de

1 Graphs - recalling basic definitions

An ***undirected graph*** $G(N, E)$ is identified by a set of nodes N and a set of edges E. Each edge (also called *undirected arc*) $e \in E$ is an unordered pair $\{i, j\}$ of distinct nodes $i, j \in N$.
A *path* is a finite sequence of nodes $i_1, i_2, \ldots, i_p$ with no repetition of nodes and such that two successive nodes in the sequence are the extreme nodes of an edge in the graph E, i.e. $\{i_\ell, i_{\ell+1}\} \in E$ for every $\ell = 1, 2, \ldots, p-1$.
A directed graph is said to be *connected* when there exists a path from i to j for every couple of distinct nodes $i, j \in N$
A *cycle* is a path such that $i_1 = i_p$ (it is common to add the requirement that the cycle contains at least 3 distinct nodes, so to exclude a cycle of the form i_1, i_2, i_1 in which an edge $\{i, j\}$ is passed through back and forth).

A ***directed graph*** $G(N, A)$ is identified by a set of nodes N and a set of arcs A. Each (directed!) arc $a \in A$ is an ordered pair (i, j) of distinct nodes $i, j \in N$.
Given a directed graph, we can derive the corresponding undirected graph by ignoring the direction of the arcs and deleting repetitions of the same arcs. A directed graph is said to be *connected* when the corresponding undirected graph is connected.
A *path* is a finite sequence of nodes $i_1, i_2, \ldots, i_p$ (with no repetition of nodes) associated with a sequence of arcs $a_1, a_2, \ldots, a_p$ such that it holds either $a_\ell = (i_\ell, i_{\ell+1})$ (forward arc) or $a_\ell = (i_{\ell+1}, i_\ell)$ (backward arc).
A *cycle* is a path such that $i_1 = i_p$. In contrast to the definition of path in undirected graph, for the directed graph case we allow to have a path made up of only two nodes (in this case, the path is of the form $i, (i, j), j, (j, i), i$.
Remark: path and cycles made up only of forward arcs are called *directed*.

A ***tree*** is a connected undirected graph $G(N, E)$ with no cycles.
We can formalize some important properties of a tree in a theorem.
Theorem 1:

1. Every tree made up of more than one node has at least one leaf.
2. An undirected graph is a tree if and only if it is connected and possesses $|N| - 1$ edges.
3. Given any two distinct nodes i, j of a tree, there exists a unique path from i to j.
4. If we add a single edge to a tree, the resulting graph contains exactly one cycle (if we do not distinguish between cycles defined over the same set of nodes).

Given an undirected graph $G(N, E)$, we call ***spanning tree*** a tree such that $G(N, E_1)$ with $E_1 \subseteq E$.
We can formalize some important properties of a tree in a theorem.
Theorem 2: Let $G(N, E)$ be a connected undirected graph and define the subset $F \subseteq E$. If the edges in F do not form any cycle, then the F can be extended to a subset F_1 such that $F \subseteq F_1 \subseteq E$ and $G(N, F_1)$ is a spanning tree.

2 Network Flow Problems

Definition (Network): A network is a directed graph $G(N, A)$ where i) each arc $a = (i, j) \in A$ has a capacity $u_{ij} \geq 0$ and is associated with a flow $f_{ij} \geq 0$ passing through the arc; ii) sending one unit of flow over an arc (i, j) entails a cost $c_{ij} \geq 0$; iii) each node is associated with a number $b_i \mathbb{R}$ representing the flow entering or leaving the network in i (in particular, if $b_i > 0$ then the node is a source and if $b_i < 0$ then the node is a sink.

A **flow** is any vector f_{ij}, $(i,j) \in A$. A **feasible flow** is a flow that additionally satisfies the following conditions:

$$\sum_{(i,j)\in A} f_{ji} - \sum_{(j,i)\in A} f_{ij} = b_i \qquad \forall i \in N \tag{1}$$

$$0 \leq f_{ij} \leq u_{ij} \qquad \forall (i,j) \in A\,. \tag{2}$$

The first condition imposes the conservation of flows in a node: the amount of flow that enters a node must be equal to the amount of flow that exits from the node. The second condition imposes that the flow on each arc must be non-negative and must satisfy the capacity limit.

Remark: summing both sides of equalities (1) over all the nodes of the graph, we obtain $\sum_{i\in N} b_i = 0$, meaning that the total flow entering the network must equal the flow exiting. This is a condition of existence of a feasible vector that we will always assume to be met in all network problems that we will consider.

In an optimization perspective, we want to find a feasible flow that minimizes the objective function $\sum_{(i,j)\in A} c_{ij} f_{ij}$ considering the cost of sending a flow over the network.

Assuming that the directed graph $G(N, A)$ of the network is such that $|N| = n$ and $|A| = m$, if we use a matrix form, the flow conservation constraints (1) can be written as:

$$Af = b$$

where f is a flow vector and A is the node-arc incidence matrix $\{-1, 0, 1\}^{n\times m}$ defined in the following way:

$$a_{ie} = \begin{cases} +1 & \text{if } i \text{ is the start node of arc } e \\ -1 & \text{if } i \text{ is the end node of arc } e \\ 0 & \text{otherwise} \end{cases}$$

Remark: Every column of the matrix A contains one $+1$ and one -1, whereas all the other entries are zero. Additionally, the sum of all the rows of A is equal to the zero vector, thus indicating that the rows of A are linearly dependent.

A ***circulation*** is a (feasible or infeasible) flow vector f such that $Af = 0$. Since $b = 0$, it denotes a flow that "circulates" inside the network and there is no flow entering or exiting the network.
Let C be a cycle and C^F, C^B be the sets of forward and backward arcs of C, respectively. The flow vector f^C defined in the following way:

$$f^C_{ij} = \begin{cases} +1 & \text{if } (i,j) \in C^F \\ -1 & \text{if } (i,j) \in C^B \\ 0 & \text{otherwise} \end{cases}$$

is called ***simple circulation*** associated with the cycle C.
Remark: A simple circulation f^C is such that $Af^C = 0$.

2.1 Uncapacitated Network Flow Problems

In this section, we consider the following network design problem, where we have dropped the capacity constraints (the design problem is then called *uncapacitated*):

$$min \quad c'f \tag{3}$$

$$Af = b \tag{4}$$

$$f \geq 0 \tag{5}$$

where A is the node-arc incidence matrix of the directed graph $G(V, A)$ representing the network.
Throughout the section, we assume that:

- the graph G is connected
- $\sum_{i \in N} b_i = 0$ (to guarantee the feasibility of the problem)

As we have previously noted, by summing the rows of the matrix A we obtain the zero vector, thus revealing that the rows of A are linearly dependant. As a consequence, we can express one row of A as a linear combination of the remaining rows of A according to coefficients that are not all simultaneously null. In particular, we can delete the last constraint of $Af = b$ (i.e., the flow conservation constraint of node n) and the set of feasible solutions does not vary. Indeed:

$$\sum_{i \in N} a'_i = 0 \Longrightarrow a'_n = \sum_{i \in N \setminus \{n\}} -a'_i$$

We can then define the *truncated node-arc incidence matrix* $\bar{A}$, obtained by deleting the last row of A corresponding to node n, and the truncated vector $\bar{b}$, obtained by deleting the last element b_n.

After having introduced the truncated matrix $\bar{A}$, we can provide a central definition.

Definition (feasible tree solution)): A flow vector f over a network $G(N, A)$i s called a tree solution if it can be defined in the following way:

1. select a set $T \subset A$: $|T| = n - 1$ that define a tree when the direction is ignored;
2. set $f_{ij} = 0 \ \forall (i, j) \notin T$;
3. determine the flow variables $f_{ij} \ \forall (i, j) \in T$ on the basis of the flow conservation constraints $\bar{A}f = \bar{b}$.

When a tree solution satisfies $f_{ij} \geq 0$, it is called a feasible tree solution.

We can prove that, once that a tree is fixed in $G(N, A)$, the corresponding tree solution is uniquely determined.

Theorem 3: Let $T \subseteq A$ be a set of cardinality $n - 1$ that defines a tree in $G(N, A)$ when the direction is ignored. Then the linear system $\bar{A}f = \bar{b}$ with $f_{ij} = 0 \ \forall (i, j) \notin T$ admits a unique solution.

Proof: Let B be the matrix of dimension $(n - 1) \times (n - 1)$ obtained from $\bar{A}$ keeping only the columns corresponding to arcs in T and let f^T be the flow (n-1)-dimensional vector made up of the flow variables $f_{ij} : (i, j) \in T$. In order to show that the system $\bar{A}f = \bar{b}$ has a unique solution, we show that B is non-singular.

As first step, we renumber the nodes so that the number increases on the path from any leaf to the root node n. Moreover, we assign the etiquette $\min\{i, j\}$ to each arc $(i, j) \in T$. This renumbering has the effect of rearranging the order of the rows and columns of $\bar{A}$, however, without changing the nature of B in terms of (non)-singularity.

Given the previous renumbering, the i-th column of B corresponds with the i-th arc, which has the form (i, j) or (j, i) with $j > i$. Since $j > i$, there are no non-zero elements in the rows above the diagonal. Additionally, since the only non-zero entries in the i-th column are i and j, B is lower triangular and there are no zero entries in the diagonal. As a consequence, the matrix B has a non-zero determinant and is thus non-singular, thus completing the proof.

Corollary: If the graph $G(N, A)$ is connected, then the truncated node-arc incidence matrix $\bar{A}$ has linearly independent rows.

Proof: Thanks to Theorem 2, we know that if a graph G is connected, then we can identify a subset of arcs T that define a tree when their direction is neglected. Given such a subset T and defined the corresponding $(n-1) \times (n-1)$ matrix B, we know from the proof of the previous theorem that B is non-singular. Therefore, the $(n - 1)$ rows of $\bar{A}$ are linearly independent.

Theorem 4: A flow vector is a tree solution if and only if it is a basic solution.

Proof: Let f be a tree solution. We can note that the columns of $\bar{A}$ corresponding to the variables f_{ij} with $(i, j) \in T$ are the (linearly independent) columns of B and, by linear programming terminology, B is thus a basis matrix. Since we set $f_{ij} = 0$ for $(i, j) \notin T$, the flow vector f is the basic solution corresponding with the basis B. This proves that a tree solution is a basic solution.

To prove that a basic solution is a tree solution, we proceed by showing that a flow vector f that is not a tree solution cannot be a basic solution. As first step, we can note that if $Af \neq b$, then f is not a basic solution by definition. As a consequence, we can focus on the case of f such that $Af = b$.

Given f: $Af = b$, define the subset F of arcs on which a non-zero flow is present (i.e., $F = \{(i, j) \in A : f_{ij} \neq 0\}$.

If the arcs of F do not define a cycle, then there exists a subset $T \subseteq F$: $|T| = n-1$ and such that T forms a tree. As $f_{ij} = 0$, $\forall (i, j) \notin T$, f is the tree solution associated with T, fact that contradicts our assumption.

Assume instead that the arcs of F defines a cycle C. Let f^C be the simple circulation associated with C. If we introduce the flow vector $f + f^C$, we have $A(f + f^C) = b$, since $Af^C = 0$ by definition of circulation. Additionally, when $f_{ij} = 0$ the arc (i, j) does not belong to C and $f^C_{ij} = 0$. We can then note that all the constraints that are active in f are also active in $f + f^C$ and the corresponding system of equations does not admit a unique solution is f is therefore not a basic solution.