4 The Shortest Path Problem

Consider a directed graph $G = (V, A)$. Given an arc $a = (i, j) \in A$, denote with $a^- = i$ the arc's origin and with $a^+ = j$ its destination. As usual, given a vertex $i \in V$, we use the following notation:

$$\delta^+(i) = \{j \in V \ : \ (i, j) \in A\}$$
$$\delta^-(i) = \{j \in V \ : \ (j, i) \in A\}.$$

A path P in G is a sequence of arcs $(a_1, \ldots, a_\ell)$ (with $a_k \in A \ \forall k \in \{1, \ldots, \ell\}$) such that $a^+_k = a^-_{k+1}$ for all $k \in \{1, \ldots, \ell - 1\}$. We call a^-_1 the origin and a^+_ℓ the destination of P. A path P is called a cycle if $a^+_\ell = a^-_1$. Graph G is called acyclic if it does not allow any cycles.

We will focus our attention on directed acyclic graphs (DAGs). Let $G = (V, A)$ be a DAG and let $s \in V$ be a source vertex and $t \in V \setminus \{s\}$ a sink vertex. Assume that $\delta^-(s) = \delta^+(t) = \emptyset$. Let $c_{ij} \geq 0$ be a cost associated with each arc $(i, j) \in A$. Given a path P, its cost c_P is the sum of the costs of the arcs making up path P.

The Shortest Path Problem (SPP) asks to find a minimal-cost path from s to t. An integer programme for this problem uses binary variables $x_{ij} \in \{0, 1\}$ for each arc $(i, j) \in A$ with the following meaning:

$$x_{ij} = \begin{cases} 1 & \text{if arc } (i, j) \text{ is part of the minimal-cost path from } s \text{ to } t \\ 0 & \text{otherwise.} \end{cases}$$

The formulation reads as follows.

$$\min \quad \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{16}$$

$$\text{subject to} \quad \sum_{j\in\delta^+(s)} x_{sj} = 1 \tag{17}$$

$$\sum_{i\in\delta^-(t)} x_{it} = 1 \tag{18}$$

$$\sum_{j\in\delta^+(i)} x_{ij} = \sum_{j\in\delta^-(i)} x_{ji} \qquad \forall i \in V \setminus \{s, t\} \tag{19}$$

$$\sum_{j\in\delta^+(i)} x_{ij} \leq 1 \qquad \forall i \in V \tag{20}$$

$$x_{ij} \in \{0, 1\} \qquad \forall (i, j) \in A. \tag{21}$$

Now consider a generalisation of the (SPP) in which the graph G can contain cycles, and the costs can be negative ($c_{ij} \in \mathbb{R}$ for all $(i, j) \in A$). Formulation (16)–(21) is no longer valid. Show an example of a graph in which the above formulation produces an optimal solution that is *not* a path from source to sink. (Hint: devise a graph containing a negative cost cycle.)

We might think to extend formulation (16)–(21) with subtour elimination constraints (SECs) similar to those we devise for the Travelling Salesman Problem (TSP):

$$\sum_{i\in S} \sum_{\substack{j\in V\setminus S \\ (i,j)\in A}} x_{ij} \geq 1 \quad \forall S \subset V \ : \ s \in S \text{ and } t \in V \setminus S. \tag{22}$$

Constraint (22) is an "obvious" adaptation of the TSP SEC, but it is wrong (as it often happens when something looks too obvious). Show an example of a graph in which constraint (22) does not prevent a subtour from forming. A correct SEC for the SPP is the following:

$$\sum_{i\in S} \sum_{\substack{j\in S \\ (i,j)\in A}} x_{ij} \leq |S| - 1 \quad \forall S \subset V \setminus \{s, t\} \text{ and } |S| \geq 2. \tag{23}$$

The above SECs use sets S that contain neither the source nor the sink and have a size of at least 2 (otherwise, there is no arc all internal to the set). Remark that (23) are similar to the alternative version of the TSP SECs.

Write a branch-and-cut algorithm to solve formulation (16)–(21), (23). You shall devise a separation procedure for (23) that works on integer and fractional solutions. Remark: Unlike the TSP, the SPP does not require all vertices of G to be "visited". Test your procedure on complete graphs (that, therefore, contain cycles) and include graphs with negative-cost cycles in your test set.