ALGORITHMS TO SOLVE SOME CLASSES OF NETWORK MINIMAX PROBLEMS AND THEIR APPLICATIONS

D. D. Lozovanu

We consider some classes of network minimax problems that generalize combinatorial versions of shortest and longest path problems in networks. Effective polynomial-time algorithms are proposed for solving these problems.

The paper considers network minimax problems which generalize in various directions some network transportation models [1] and find applications in the analysis and solution of cyclic games [2]. The main focus is on choosing a minimax path in an acyclic transportation network, which is a generalization of the combinatorial problem of finding the shortest and longest paths in weighted directed acyclic graphs. We show that effective strongly polynomial algorithms can be developed to find the minimax path in an acyclic network, and these algorithms are very useful for the analysis and solution of some known classes of problems. Thus, the algorithms proposed in this paper and the results of [2] are applied to prove the existence of polynomial-time algorithms for the minimax mean cycle problem considered in [2]. We also show that these algorithms may be applied directly in real-life network transportation models.

1. The minimax path problem formulated in this section may be considered for general, not necessarily acyclic, networks. However, in general, this problem is not always solvable, and we will therefore consider it only for acyclic networks.

Given is a network $G = (V, E)$, $|V| = n$, $|E| = m$, in the form of a directed acyclic graph with the sink $v_t \in V$. The function $c : E \to \mathbb{R}$ is defined on the set of arcs $E$; this function is called the arc cost function of the network $G$. We assume that a shipment of unit mass is located at some vertex $v_0 \in V$ and has to be transported from $v_0$ to $v_t$. If the objective function is to minimize the transportation costs in the process of shipping, we have to choose a trajectory along a path of minimum total arc cost.

This problem can be solved by numerous algorithms with sufficiently good strongly polynomial bounds [3, 4]. Some of these algorithms may be used to solve the problem in a more general form, specifically to find the shortest path tree $T^* = (V, E^*)$ from any other vertex $v \in V \setminus \{v_t\}$ to $v_t$. Thus, regardless of the particular vertex where the shipment is located, the optimal shipping strategy to $v_t$ can be based on the following rule: successively pass from one vertex of the graph $G$ to another, choosing in each step a path along the arcs $e \in E^*$, until $v_t$ is reached.

Note that if $V_G(v)$ is the set of vertices that are one-step reachable in the network $G$ from the vertex $v$ by following the arcs out of $v$, then the choice

$$s^* : v \to u \in V_{T^*}(v) \text{ for } v \in V \setminus \{v_t\}$$

is unique. Therefore, if $T = (V, E)$ is an arbitrary tree with the sink $v_t$ in $G$, then the mapping

$$s : v \to u \in V_T(v) \text{ for } v \in V \setminus \{v_t\}$$

may be interpreted as a stationary strategy of motion in the given problem, and the mapping $s^*$ is the optimal stationary shipping strategy. We see that to each strategy $s$ in $G$ corresponds some tree $T_s = (V, E_s)$ and conversely to any tree $T = (V, E)$ corresponds some strategy $s_T$ in $G$.
Now, as in [1], we assume that the vertex set \( V \) of the network is partitioned into two nonintersecting subsets \( V_A \) and \( V_B \) such that \( V_A \cap V_B = \emptyset \) so that the next transition \( u \in V_G(v) \) from the vertex \( v \) in the given shipping process can be chosen only if \( v \) is contained in the set \( V_A \subseteq V \). Then using the guaranteed payoff paradigm, we assume that the transition \( u \in V_G(v) \) for the vertices \( v \in V_B \) is chosen by the adversary, and the vertices \( v \in V_A \) and \( v \in V_B \) are respectively called the positions of players A and B (who have the right of next move).

Consider the pair of mappings
\[
\begin{align*}
S_A : v &\rightarrow V_G(v) \quad \text{for} \quad v \in V_A \setminus \{v_t\}, \\
S_B : v &\rightarrow V_G(v) \quad \text{for} \quad v \in V_B \setminus \{v_t\},
\end{align*}
\]
which are respectively called the stationary strategies of players A and B, and fix the tree \( T_s = (V, E_s) \) corresponding to these strategies. For a given initial position \( v = v_0 \), the realization of these strategies produces total transportation costs \( \hat{c}(S_A, S_B, v) \) for shipping from vertex \( v \) to vertex \( v_t \) which are equal to the sum of the arc costs along the unique path in the tree \( T_s \) from \( v \) to \( v_t \). The function \( \hat{c}(S_A, S_B, v) \) defined in this way in the Cartesian product of the sets of stationary strategies \( S_A \) and \( S_B \) essentially defines a finite game, which we call an acyclic c-game. An acyclic c-game can be defined by specifying the game network \( (G, V_A, V_B, c) \) and the initial position \( v = v_0 \).

As we shall see below, for any initial position \( v \) we have discrete identity of the minimax and the maximin,
\[
\min_{S_A} \max_{S_B} \hat{c}(S_A, S_B, v) = \max_{S_B} \min_{S_A} \hat{c}(S_A, S_B, v) = p(v),
\]
which implies the existence of optimal stationary strategies of players A and B in the given finite acyclic c-game.

To prove this assertion, we consider the potential transformations \( c'(v, u) = c(v, u) + \varepsilon(u) - \varepsilon(v) \) of the network arc costs, where \( \varepsilon : V \rightarrow \mathbb{R} \) is an arbitrary function defined on the network vertices (we call it potential). It is easy to show that the potential transformation preserves the structure of the optimal stationary strategies in this finite game, i.e., the structure of the sought tree \( T_s = (V, E_s) \). Therefore, the function \( c' \) produced by potential transformations will be called equivalent to \( c \). Note that contrary to cyclic games [2], the values \( p(v) \) of the game positions may change under equivalent transformations.

Assume that the acyclic game network \( (G, V_A, V_B, c) \) has the following properties:
(i) for any vertex \( v \in V_A \setminus \{v_t\} \), all the outgoing arcs \( e = (v, u) \in E \) have nonnegative costs;
(ii) for any vertex \( v \in V_B \setminus \{v_t\} \), all the outgoing arcs \( e = (v, u) \in E \) have nonpositive costs;
(iii) for an arbitrary vertex \( v \in V \setminus \{v_t\} \) there exists at least one arc \( e = (v, u) \) leaving \( v \) for which \( c(v, u) = 0 \).

It is easy to see that if the game network \( (G, V_A, V_B, c) \) satisfies conditions (i)-(iii), then the value \( p(v) \) of any position \( v \in V \) is zero and the search for optimal stationary strategies of the players in \( G \) becomes trivial: in each move, both players select a zero-cost arc among the arcs leaving the current positions. Such networks are called canonical game networks.

**Theorem 1.** For an arbitrary acyclic game network \( (G, V_A, V_B, c) \) there exists an equivalent potential transformation \( c' \) of the costs of the arcs \( e \in E \) which reduces an acyclic game network to canonical form.

**Proof.** Without loss of generality, we assume that the network vertices are indexed 1 to \( n \) so that if the index \( u \) of some vertex is greater than the index \( v \) of another vertex, then there is no path \( p(v, u) \) from \( v \) to \( u \) in \( G \). This vertex indexing for acyclic networks is known to be feasible [3] and it can be realized by an \( O(n^2) \) algorithm. With this vertex indexing, \( v_t \) is assigned the index 1.

For the vertex \( v \in V \) define the numbers \( \varepsilon(v) \) as follows:
\[
\varepsilon(v) = \begin{cases} 
0, & v = 1; \\
\min_{u \in V_G(v)} (\varepsilon(u) + c(v, u)), & v \in V_A; \\
\max_{u \in V_G(v)} (\varepsilon(u) + c(v, u)), & v \in V_B. 
\end{cases}
\]

This completes the proof. The numbers \( \varepsilon(v) \) essentially define an equivalent potential transformation \( c' \) which reduces an acyclic game network to canonical form.