ARTICLE IN PRESS
Discrete Applied Mathematics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

# The shortest connection game☆

Andreas Darmann [a], Ulrich Pferschy [b], Joachim Schauer [c,*]

[a] *Institute of Public Economics, University of Graz, Universitaetsstr. 15, 8010 Graz, Austria*
[b] *Department of Statistics and Operations Research, University of Graz, Universitaetsstr. 15, 8010 Graz, Austria*
[c] *Department of Mathematics, University of Klagenfurt, Universitaetsstr. 65-67, 9020 Klagenfurt, Austria*

ARTICLE INFO

*Article history:*
Received 26 November 2015
Received in revised form 10 November 2016
Accepted 23 January 2017
Available online xxxx

*Keywords:*
Shortest path problem
Game theory
Computational complexity
Cactus graph

ABSTRACT

We introduce Shortest Connection Game, a two-player game played on a directed graph with edge costs. Given two designated vertices in which the players start, the players take turns in choosing edges emanating from the vertex they are currently located at. This way, each of the players forms a path that origins from its respective starting vertex. The game ends as soon as the two paths meet, i.e., a connection between the players is established. Each player has to carry the cost of its chosen edges and thus aims at minimizing its own total cost.

In this work we analyse the computational complexity of Shortest Connection Game. On the negative side, Shortest Connection Game turns out to be computationally hard even on restricted graph classes such as bipartite, acyclic and cactus graphs. On the positive side, we can give a polynomial time algorithm for cactus graphs when the game is restricted to simple paths.



## 1. Introduction

We consider the following game on a directed graph $G = (V, E)$ with a vertex set $V$, a set of directed edges $E$ with nonnegative edge costs $c(u, v)$ for each edge $(u, v) \in E$, and two designated vertices $s, t \in V$. We assume that $G$ is connected, but not necessarily strongly connected.

In Shortest Connection Game two players $A$ and $B$ start from their respective homebases ($A$ in $s$ and $B$ in $t$). The aim of the game is to establish a connection between $s$ and $t$ in the following sense. The players take turns in moving along an edge and thus each of them constructs a directed path. The game ends as soon as one player reaches a vertex $m$, i.e. a meeting point, which was already visited by the other player. This means that at the end of the game one player, say $A$, has selected a path from $s$ to $m$, while $B$ has selected a path from $t$ to $m$ and possibly further on to additional vertices (or vice versa). It does not matter which player makes the last move. A player cannot remain at its vertex but has to move whenever it is its turn. Each player has to carry the cost of its chosen edges and wants to minimize the total costs it has to pay. A small motivational example concerning a practical decision problem can be found in an accompanying technical report [5].

Note that it is not always beneficial for both players to move closer to each other. Instead, one player may take advantage of cheap edges and move away from the other player, who then has to bear the burden of building the connection.

☆ Andreas Darmann was supported by the Austrian Science Fund (FWF): [P 23724-G11]. Ulrich Pferschy and Joachim Schauer were supported by the Austrian Science Fund (FWF): [P 23829-N13]. This research has been supported by the University of Graz project "Choice-Selection-Decision".
* Corresponding author.
*E-mail addresses:* andreas.darmann@uni-graz.at (A. Darmann), pferschy@uni-graz.at (U. Pferschy), joachim.schauer@aau.at (J. Schauer).

http://dx.doi.org/10.1016/j.dam.2017.01.024
0166-218X/© 2017 Published by Elsevier B.V.

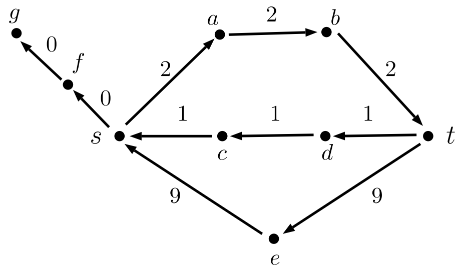


**Fig. 1.** Instance of Shortest Connection Game where $A$ can force $B$ into a solution with cost 18 for $B$ by moving into the dead end path towards $g$.

To avoid unnecessary technicalities we assume that for every graph considered a solution of the game does exist. This could be checked in a preprocessing step e.g. by executing a breadth-first-search starting in parallel from $s$ and $t$.

We will impose two restrictions on the problem setting to ensure that the two players actually meet in some vertex and to guarantee finiteness of the game. To enable a feasible outcome of the game we restrict the players in every decision to choose only edges which still permit a meeting point of the two paths (in analogy to (R1) in [6]).

(R1) The players cannot select an edge which does not permit the paths of the players to meet.

Clearly, (R1) is always satisfied if $G$ is strongly connected. Otherwise, a player might block the game by moving into a dead end street of the graph or into a region of the graph which the other the player cannot reach by a directed path. Selecting an edge a player also has to observe that the other player must be able to choose an edge in every round and does not get stuck. This aspect may lead to interesting outcomes as shown in the following example depicted in Fig. 1.

Player $A$ could move via vertices $a$ and $b$ to $t$ at a cost of 6. However, $A$ will prefer the dead end path to $g$ at zero cost. This forces player $B$ to take the expensive path via $e$ to $s$ with cost 18 because the cheaper path via $c$ and $d$ would leave player $A$ stuck in $g$ with no possibility to move in the third round.

Secondly, we want to guarantee that the game remains finite. This raises the question of how to handle cycles. It could be argued that moving in a cycle makes sense for a player who wants to avoid an edge of large costs. However, for the sake of finiteness each cycle should be traversed at most once as guaranteed by the following restriction:

(R2) Each edge may be used at most once.

Note that (R2) does not rule out the possibility to visit a vertex more than once. However, it is also interesting to restrict the game to simple paths and thus exclude cycles completely by the following condition.

(R3) Each vertex may be used at most once by each player.

While (R1) and (R2) will be strictly enforced throughout the paper, we will consider the variants with and without (R3).

In this paper we will study Shortest Connection Game on directed graphs exclusively (also the case of an undirected graph is interesting, but this is beyond the scope of the current paper). An undirected graph will appear only in Section 2.2 as an instance of Vertex Cover used in the reduction of a complexity proof. At that point we will denote undirected edges as sets $e = \{u, v\}$ while directed edges are always given as ordered pairs $(u, v)$.

We will consider general directed graphs and the following relevant special graph classes. Directed acyclic graphs do not contain a directed cycle. Directed bipartite graphs permit a partitioning of the vertex set into two subsets such that no edge has both endpoints in the same subset. Trees are directed graphs where the removal of the orientation from all edges yields an acyclic undirected graph.[1] Note that this is more general than the usual definition of a rooted, directed tree.

Finally, we consider two variants of directed cactus graphs. Recall that an undirected cactus graph is a connected graph where each edge is contained in at most one simple cycle. Equivalently, any two simple cycles have at most one vertex in common. This means that one could contract each cycle into a vertex in a unique way and obtain a tree. Cactus graphs are a subclass of series–parallel graphs and thus have treewidth at most 2. They have been used frequently in the literature as special cases of combinatorial optimization problems (see e.g. [1] or [11]). In our definition a graph is a *directed cactus graph* if the removal of the orientation from all edges yields a cactus graph in the resulting undirected graph (possibly including parallel edges). Note that also a more restrictive definition was used in the literature. Based on [15,16] we define a *strongly connected directed cactus graph* as a directed cactus graph which is strongly connected, i.e. where for any pair of vertices $u, v$, there is a directed path from $u$ to $v$. It is easy to see that strongly connected directed cactus graphs can only consist of directed cycles and any two cycles may intersect in exactly one vertex.

### *1.1. Game theoretic setting*

The game described above can be represented as a finite game in extensive form. All feasible decisions for the players can be represented in a game tree, where each node corresponds to the decision of a certain player in a vertex of the graph $G$. A similar representation was given in [6] for Shortest Path Game (see also Section 1.2).

[1] Two directed edges $(u, v)$ and $(v, u)$ imply two parallel edges $(u, v)$ and rule out the tree property.

Please cite this article in press as: A. Darmann, et al., The shortest connection game, Discrete Applied Mathematics (2017), http://dx.doi.org/10.1016/j.dam.2017.01.024