A linear-time algorithm to find a pair of arc-disjoint spanning in-arborescence and out-arborescence in a directed acyclic graph

Kristóf Bérczi\textsuperscript{a}, Satoru Fujishige\textsuperscript{b,1}, Naoyuki Kamiyama\textsuperscript{c,*,1}

\textsuperscript{a} Department of Operations Research, Eötvös Loránd University, Hungary
\textsuperscript{b} Research Institute for Mathematical Sciences, Kyoto University, Japan
\textsuperscript{c} Department of Information and System Engineering, Chuo University, Japan

1. Introduction

In this paper, we are given a directed graph $D = (V, A)$ and it is assumed that $D$ has no loop, but $D$ may have parallel arcs. Furthermore, we assume that $D$ is weakly connected, i.e., $|V| - 1 \leq |A|$ holds. For each $a \in A$, we denote by $t(a)$ and $h(a)$ the tail and the head of $a$, respectively. For each $v \in V$, we denote by $\delta_D(v)$ and $\partial_D(v)$ the set of arcs $a \in A$ with $t(a) = v$ and $h(a) = v$, respectively. An acyclic subgraph $T = (W, B)$ of $D$ is called an \textit{in-arborescence} (resp., an \textit{in-arborescence}) when there exists $r \in W$ such that $|\delta_T(r)| = 0$ (resp., $|\partial_T(r)| = 0$), and $|\delta_T(v)| = 1$ (resp., $|\partial_T(v)| = 1$) holds for each $v \in W \setminus \{r\}$. We call $T$ an \textit{r-out-arborescence} (resp., an \textit{r-in-arborescence}). Namely, an \textit{r-out-arborescence} (resp., an \textit{r-in-arborescence}) is a rooted tree in which all arcs are directed away from (resp., toward) the root $r$.

The problem of finding $k$ arc-disjoint spanning $r$-out-arborescences for a given root $r \in V$ is very important not only from the theoretical viewpoint but also from practical viewpoints, and it has been extensively studied. It is known \cite{10,9,11,5,3} that this problem can be solved in polynomial time, and several extensions have been considered in \cite{7,4,2}. However, in many situations, we have to simultaneously consider not only an in-arborescence but also an out-arborescence. For example, in evacuation situations, an in-arborescence represents roads which refugees use. On the other hand, an out-arborescence represents roads used by emergency vehicles. Unfortunately, it is known \cite{1} that the problem of finding a pair of arc-disjoint spanning $r_1$-in-arborescence and $r_2$-out-arborescence for given roots $r_1, r_2 \in V$ is $\mathcal{NP}$-complete even if $r_1 = r_2$. As a special case, it is only known \cite{1} that this problem in a tournament can be solved in polynomial time. In this paper, we consider this problem in a directed acyclic graph. We also consider an extension of our problem to the case where we have multiple roots for in-arborescences and out-arborescences, respectively.

© 2009 Elsevier B.V. All rights reserved.

Theorem 1. Given a directed acyclic graph $D = (V, A)$ with roots $r_1, r_2 \in V$, we can discern the existence of a pair of arc-
disjoint spanning $r_1$-in-arborescence and $r_2$-out-arborescence, and find such arborescences, if they exist, in $O(|A|)$ time.

The rest of this paper is organized as follows. In Section 2, we introduce a bipartite graph associated with our problem, and then we show that our problem is equivalent to the problem of finding a certain matching in this bipartite graph. In Section 3, we give a linear-time algorithm for discerning the existence of a desired matching in the bipartite graph, and for finding it if one exists. In Section 4, we consider the case where we have multiple roots for in-arborescences and out-arborescences, respectively.

2. An associated bipartite graph

In this section, we define a bipartite graph $G_D = (X, Y; E)$ associated with our problem for a directed acyclic graph $D = (V, A)$, and we show that our problem in $D$ is equivalent to the problem of finding a matching that covers all vertices of $Y$ in $G_D$. In the sequel, we assume without loss of generality that $\delta_D(r_1) = \emptyset$ and $\varrho_D(t_2) = \emptyset$ holds. Note that if $\delta_D(r_1) \neq \emptyset$ or $\varrho_D(t_2) \neq \emptyset$ holds, there exists no feasible solution since $D$ is acyclic.

2.1. Definitions

Define a bipartite graph $G_D = (X, Y; E)$ with two vertex sets $X$ and $Y$ and an edge set $E$ between $X$ and $Y$ as follows.

(i) Vertex set $X$ is given by $X = \{x(a) \mid a \in A\}$, where $|X| = |A|$.

(ii) Vertex set $Y$ consists of two disjoint sets $Y^+$ and $Y^-$ given by $Y^+ = \{y^+(v) \mid v \in V \setminus \{r_1\}\}$ and $Y^- = \{y^-(v) \mid v \in V \setminus \{r_2\}\}$.

(iii) For each $a \in A$, we have two edges in $E$: one connects $x(a)$ and $y^+(t(a))$ and the other connects $x(a)$ and $y^-(h(a))$. That is, $E = \{(x(a), y^+(t(a))) \mid a \in A\} \cup \{(x(a), y^-(h(a))) \mid a \in A\}$.

For example, for a directed graph $D$ in Fig. 1(a) the bipartite graph $G_D$ becomes the one as illustrated in Fig. 1(b).

Here we introduce notations to be used in the subsequent arguments (see Fig. 2). For each $v \in X \cup Y$, we denote by $\deg_{G_D}(v)$ the degree of $v$ in $G_D$. For each $e \in E$, let $\partial_X(e)$ (resp., $\partial_Y(e)$) be the endpoint of $e$ belonging to $X$ (resp., $Y$). For each $e \in E$, we denote by $p(e)$ the edge $e' \in E$ with $e \neq e'$ and $\partial_X(e) = \partial_X(e')$. Notice that since $\deg_{G_D}(x) = 2$ holds for each $x \in X$ by the definition of $G_D$, $p(e)$ is uniquely determined for each $e \in E$.

2.2. An equivalent problem on a bipartite graph

In this subsection, we show the equivalence between our problem for $D$ and the problem of finding a matching in $G_D$ which covers all vertices of $Y$.

Lemma 2. Given a directed acyclic graph $D = (V, A)$ with roots $r_1, r_2 \in V$, there exists a pair of arc-disjoint spanning $r_1$-in-arborescence $T_1$ and $r_2$-out-arborescence $T_2$ if and only if there exists a matching $M$ in $G_D = (X, Y; E)$ which covers all vertices of $Y$.

Proof. Since it is not difficult to see the “only if” part of the lemma, we show the “if” part. Let $M$ be a matching in $G_D$ which covers all vertices of $Y$. Let $A^+$ (resp., $A^-$) be the set of arcs $a \in A$ such that $x(a)$ is connected with some vertex of $Y^+$ (resp., $Y^-$) by an edge of $M$. Let $T_1$ (resp., $T_2$) be the subgraph $(V, A^+)$ (resp., $(V, A^-)$) of $D$. Since $M$ covers all vertices of $Y$, $|\partial_T(v)| = 1$ (resp., $|\partial_{T_2}(v)| = 1$) holds for each $v \in V \setminus \{r_1\}$ (resp., $V \setminus \{r_2\}$). Thus, since $D$ is acyclic, $T_1$ and $T_2$ are a spanning $r_1$-in-arborescence and a spanning $r_2$-out-arborescence, respectively. Furthermore, since $M$ is a matching, $A^+$ and $A^-$ are disjoint, which implies $T_1$ and $T_2$ are arc-disjoint. This completes the proof of the “if” part.

The latter half of the lemma immediately follows from the proof of the “if” part. □

By Lemma 2, we can discern the existence of a pair of arc-disjoint spanning $r_1$-in-arborescence and $r_2$-out-arborescence, and find such arborescences if they exist, by computing a maximum matching of $G_D$. Hence, we can solve our problem in polynomial time by using bipartite-matching algorithms such as in [6]. However, we show in the subsequent section that we can discern the existence of a matching of $G_D$ which covers all vertices of $Y$ and find such a matching if one exists, in $O(|A|)$ time.
3. A linear-time algorithm

The goal of this section is to show the following theorem, which implies Theorem 1 by Lemma 2.

Theorem 3. Given a directed acyclic graph $D = (V, A)$ with roots $r_1, r_2 \in V$, we can discern the existence of a matching in $G_D = (X, Y; E)$ which covers all vertices of $Y$ and find such a matching if one exists, in $O(|A|)$ time.

In the subsequent arguments, we assume without loss of generality that $\deg_{G_D}(y) \geq 1$ holds for every $y \in Y$ since if there exists a vertex $y \in Y$ with $\deg_{G_D}(y) = 0$, there exists no solution. We divide the proof into two parts corresponding to the following two cases.

Case 1: For every $y \in Y$, $\deg_{G_D}(y) \geq 2$ holds. Case 2: There exists a vertex $y \in Y$ with $\deg_{G_D}(y) = 1$.

We first show that in Case 1, there always exists a matching in $G_D$ which covers all vertices of $Y$, and we can find such a matching in $O(|A|)$ time. Then, we show that in Case 2, we can discern the existence of a matching in $G_D$ which covers all vertices of $Y$, and reduce the problem to Case 1 if any such matching exists, in $O(|A|)$ time.

3.1. Case 1

We prove the following lemma for Case 1.

Lemma 4. Given a directed acyclic graph $D = (V, A)$ with roots $r_1, r_2 \in V$, if $\deg_{G_D}(y) \geq 2$ holds for every $y \in Y$, there always exists a matching in $G_D = (X, Y; E)$ which covers all vertices of $Y$, and we can find one such matching in $O(|A|)$ time.

Proof. Let $\tilde{G}_D = (X \cup \{s\}, Y; \tilde{E})$ be the bipartite graph obtained from $G_D$ by adding a new vertex $s$ and connecting edges between $s$ and each odd-degree vertex $y \in Y$ (see Fig. 3(a)). It is easy to see that $|E| \leq |E| + |Y| = |E| + 2(|V| - 1)$. Furthermore, since $\deg_{G_D}(x) = 2$ holds for every $x \in X$, we have $|E| = 2|X| = 2|A|$. Hence, $|\tilde{E}| = O(|A|)$ holds, and our goal is to find a desired matching in $O(|\tilde{E}|)$ time.

Since the sum of the degrees of all vertices $x \in X$ is even, the degree of $s$ in $\tilde{G}_D$ is even. This implies that $\tilde{G}_D$ is an Eulerian graph. Hence, $\tilde{G}_D$ consists of several edge-disjoint cycles (see Fig. 3(b)), which can be computed in $O(|\tilde{E}|)$ time by using an algorithm for finding Eulerian walk [for a standard algorithm, see [8]]. Let $M$ be the set of edges of $\tilde{G}_D$ obtained from all the cycles by choosing every other edge along the cycles (see Fig. 3(b)). Then every vertex $v$ of $\tilde{G}_D$ has $\deg_{\tilde{G}_D}(v) \geq 4$, so such a vertex $v$ is incident to at least two edges in $\tilde{M}$. Hence, letting $M = \tilde{M} \cap E$, $M$ satisfies the following conditions. (Note that $M$ is obtained by removing from $\tilde{M}$ the edges incident to $s$ in $\tilde{G}_D$.)

A1. $M$ covers all vertices of $\tilde{Y}$.
A2. Each $x \in X$ is covered by exactly one edge in $M$.

By conditions A1 and A2, we can obtain a matching in $G_D$ which covers all vertices of $Y$ by appropriately removing edges from $M$. This completes the proof.

3.2. Case 2

In this subsection, we show that in Case 2 we can discern the existence of a feasible solution of our problem and reduce the problem to Case 1 if one exists, in $O(|A|)$ time. This will complete the proof of Theorem 3.

The following lemma asserts that we can reduce Case 2 to Case 1 by greedily removing vertices with degree one.

Lemma 5. Suppose that we are given a directed acyclic graph $D = (V, A)$ with roots $r_1, r_2 \in V$, and a vertex $\tilde{y} \in Y$ with $\deg_{G_D}(\tilde{y}) = 1$, denoting by $\tilde{e} \in E$ the single edge incident to $\tilde{y}$. Let $\tilde{G}_D = (\tilde{X}, \tilde{Y}; \tilde{E})$ be the bipartite graph obtained from $G_D = (X, Y; E)$ by removing vertices $\tilde{y}$ and $\partial_{\tilde{X}}(\tilde{e})$ and edges $\tilde{e}$ and $p(\tilde{e})$ (see Fig. 4). Then, there exists a matching $M$ in $\tilde{G}_D$ which covers all vertices of $\tilde{Y}$ if and only if there exists a matching $\tilde{M}$ in $\tilde{G}_D$ which covers all vertices of $\tilde{Y}$.

Proof. We first prove the “if” part. Assume that there exists a matching $M$ in $\tilde{G}_D$ which covers all vertices of $\tilde{Y}$.
Then, we can construct a matching $M$ in $G_D$ which covers all vertices of $Y$ by adding $\bar{e}$ to $M$.

Next we prove the “only if” part. Assume that there exists a matching $M$ in $G_D$ which covers all vertices of $Y$. Since $\text{deg}_{G_D}(\bar{e}) = 1$, $\bar{e}$ must be included in $M$, and $p(\bar{e})$ is not included in $M$. Hence, we can construct a matching $M$ in $G_D$ which covers all vertices of $Y$ by removing $\bar{e}$ from $M$. □

By Lemma 5, we can describe the procedure in which we can discern the existence of a feasible solution of our problem, and reduce the problem to Case 1 if one exists, in $O(|A|)$ time as in Procedure 1.

It should be noted that since $Q$ contains all vertices $y \in Y$ with $\text{deg}_{G_D}(y) = 1$ in each iteration of Step 3, the procedure is correct. Furthermore, we can easily see the following lemma, due to Lemma 5.

**Lemma 6.** Given a directed acyclic graph $D = (V, A)$ with roots $r_1, r_2 \in V$, Procedure 1 always terminates in $O(|A|)$ time. Suppose that Procedure 1 returns a bipartite graph $G'_D = (X', Y'; E')$ and a matching $M_0$. Then, we have $\text{deg}_{G'_D}(x) = 2$ for every $x \in X'$ and $\text{deg}_{G'_D}(y) \neq 1$ for every $y \in Y'$. If there exists a vertex $y$ in $G'_D$ such that $\text{deg}_{G'_D}(y) = 0$, then there does not exist any pair of arc-disjoint spanning $r_1$-in-arborescence and $r_2$-out-arborescence. Otherwise we can construct a matching $M$ in $G_D$ which covers all vertices of $Y$, from a matching $M'$ in $G'_D$ which covers all vertices of $Y'$, by putting $M \leftarrow M' \cup M_0$.

### 3.3. A full description of our algorithm

We are now ready to describe a linear-time algorithm for our problem.

1. If there exists $y \in Y$ with $\text{deg}_{G_D}(y) = 1$, apply Procedure 1 and let $G'_D$ and $M_0$ be the output of Procedure 1. If there exists a vertex whose degree is equal to zero in $G'_D$, return NULL (there exists no feasible solution). Otherwise, put $G_D \leftarrow G'_D$ and go to Step 2.
2. Find a matching $M$ in $G_D$ covering all vertices of $Y$ as described in the proof of Lemma 4, and put $M \leftarrow M \cup M_0$.
3. Using the matching $M$ in $G_D$, compute a pair of arc-disjoint spanning $r_1$-in-arborescence $T_1$ and $r_2$-out-arborescence $T_2$ and return $T_1$ and $T_2$.

It follows from Lemmas 4 and 6 that the above algorithm can find a matching in $G_D$ which covers all vertices of $Y$ if one exists in $O(|A|)$ time. This completes the proof of Theorem 3.

### 4. An extension to multiple roots

In this section, we consider the case where we have multiple roots for in-arborescences and out-arborescences, respectively. Suppose that we are given a directed acyclic graph $D = (V, A)$, two disjoint finite index sets $I_1$ and $I_2$, and a root $r_i \in V$ for each $i \in I_1 \cup I_2$, where we allow $r_i = r_j$ for distinct $i, j$. We assume without loss of generality that $\delta_D(r_i) = \emptyset$ (resp., $\delta_D(r_i) = \emptyset$) holds for each $i \in I_1$ (resp., $i \in I_2$). Let $R_1$ (resp., $R_2$) be the set $\{r_i | i \in I_1\}$ (resp., $\{r_i | i \in I_2\}$). Then we consider the problem of discerning the existence of a set of arc-disjoint $r_i$-in-arborescences $T_i$ ($i \in I_1$) and $r_i$-out-arborescences $T_i$ ($i \in I_2$) such that for each $i \in I_1$ (resp., $i \in I_2$) the vertex set of $T_i$ is $(V \setminus R_1) \cup \{r_i\}$ (resp., $(V \setminus R_2) \cup \{r_i\}$). In the same manner as in Section 2, we can see that there exist desired arborescences if and only if there exists a matching which covers all vertices of $Y$ in a bipartite graph $G_D = (X, Y; E)$ defined as follows.

(i') Vertex set $X$ is given by $X = \{x(a) | a \in A\}$, where $|X| = |A|$. 

(ii') Vertex set $Y$ consists of disjoint sets $Y_i^+ (i \in I_1)$ and $Y_i^- (i \in I_2)$. For each $i \in I_1$ (resp., $i \in I_2$), $Y_i^+$ (resp., $Y_i^-$) is given by $\{y_i^+(v) | v \in V \setminus R_1\}$ (resp., $\{y_i^-(v) | v \in V \setminus R_2\}$).

(iii') The edge set $E$ consists of two sets $E^+$ and $E^-$. For each $a \in A$ with $h(a) \notin R_1$ (resp., $t(a) \notin R_2$) and $i \in I_1$ (resp., $i \in I_2$), we connect $x(a)$ and $y_i^+(t(a))$ (resp., $y_i^-(h(a))$) by an edge in $E^+$ (resp., $E^-$). For each $a \in A$ with $h(a) \in R_1$ (resp., $t(a) \in R_2$), we connect $x(a)$ and $y_i^+(t(a))$ (resp., $y_i^-(h(a))$) for $i \in I_1$ with $h(a) = r_i$ (resp., $i \in I_2$ with $t(a) = r_i$). The edge sets $E^+$ and $E^-$ contain no other edge.

We can discard the existence of desired arborescences and find them if they exist, by computing a maximum matching in $G_D$. However, notice that $\text{deg}_{G_D}(x) \geq 3$ may hold for each $x \in X$, which is different from the case of the problem of finding a pair of an in-arborescence and an out-arborescence. It is left open whether we can find desired arborescences more efficiently than by using existing bipartite matching algorithms.

**Acknowledgements**

We are very grateful to András Frank for his useful discussions on the present topics and to the referees for their careful reading.

**References**

