Title	Covering Directed Graphs by In-trees (Mathematical Programming in the 21st Century : Optimization Modeling and Algorithms)
Author(s)	Kamiyama, Naoyuki; Katoh, Naoki
Citation	数理解析研究所講究録 (2009), 1629: 8-14
Issue Date	2009-02
URL	http://hdl.handle.net/2433/140375
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

数理解析研究所講究録
第 1629 巻 2009 年 8-14

内向木による有向グラフの被覆
Covering Directed Graphs by In-trees

神山 直之[1], 加藤 直樹[2]
Naoyuki Kamiyama, Naoki Katoh
京都大学大学院工学研究科建築学専攻
Department of Architecture and Architectural Engineering, Kyoto University

Abstract

Given a directed graph $D=(V,A)$ with a set of d specified vertices $S=\{s_1,\ldots,s_d\}\subseteq V$ and a function $f\colon S\rightarrow\mathbb{Z}_+$ where $\mathbb{Z}_+$ denotes the set of non-negative integers, we consider the problem which asks whether there exist $\sum_{i=1}^{d}f(s_i)$ in-trees denoted by $T_{i,1},T_{i,2},\ldots,T_{i,f(s_i)}$ for every $i=1,\ldots,d$ such that $T_{i,1},\ldots,T_{i,f(s_i)}$ are rooted at s_i, each $T_{i,j}$ spans vertices from which s_i is reachable and the union of all arc sets of $T_{i,j}$ for $i=1,\ldots,d$ and $j=1,\ldots,f(s_i)$ covers A. In this paper, we prove that such set of in-trees covering A can be found in time bounded by a polynomial in $\sum_{i=1}^{d}f(s_i)$ and the size of D.

1 Introduction

The problem for covering a graph by subgraphs with specified properties (for example, trees or paths) is very important from practical and theoretical viewpoints and have been extensively studied. For example, Nagamochi and Okada [6] studied the problem for covering a set of vertices of a given undirected tree by subtrees, and Arkin et al. [1] studied the problem for covering a set of vertices or edges of a given undirected graph by subtrees or paths. These results were motivated by vehicle routing problems. Moreover, Even et al. [2] studied the covering problem motivated by nurse station location problems.

This paper studies the problem for covering a directed graph by rooted trees which is motivated by the following evacuation planning problem. Given a directed graph which models a city, vertices model intersections and buildings, and arcs model roads connecting these intersections and buildings. People exist not only at vertices but also along arcs. Suppose we have to give several evacuation instructions for evacuating all people to some safety place. In order to avoid disorderly confusion, it is desirable that one evacuation instruction gives a single evacuation path for each person and these paths do not cross each other. Thus, we want each evacuation instruction to become an in-tree rooted at some safety place. Moreover, the number of instructions for each safety place is bounded in proportion to the size of each safety place.

The above evacuation planning problem is formulated as the following covering problem defined on a directed graph. We are given a directed graph $D=(V,A,S,f)$ which consists of a vertex set V, an arc set A, a set of d specified vertices $S=\{s_1,\ldots,s_d\}\subseteq V$ and a function $f\colon S\rightarrow\mathbb{Z}_+$ where $\mathbb{Z}_+$ denotes the set of non-negative integers. In the above evacuation planning problem, S corresponds to a set of safety places, and $f(s_i)$ represents the upper bound of the number of evacuation instructions for $s_i\in S$. For each $i=1,\ldots,d$, we define $V_D^i\subseteq V$ as the set of vertices in V from which s_i is reachable in D, and we define an in-tree rooted at s_i which spans V_D^i as a (D,s_i)*-in-tree*. Here an in-tree is a subgraph T of D such that T has no cycle when the direction of an arc is ignored and all arcs in T is directed to a root. We define a set $\mathcal{T}$ of $\sum_{i=1}^{d}f(s_i)$ subgraphs of D as a *D-feasible set of in-trees* if $\mathcal{T}$ contains exactly $f(s_i)$ (D,s_i)-in-trees for every $i=1,\ldots,d$. If every two distinct in-trees of a D-feasible set $\mathcal{T}$ of in-trees are

---

[1] Supported by JSPS Research Fellowships for Young Scientists.
[2] Supported by a Grant-in-Aid for Scientific Research (C), JSPS.

arc-disjoint, we call $\mathcal{T}$ a *D-feasible set of arc-disjoint in-trees.* Furthermore, if the union of arc sets of all in-trees of a D-feasible set $\mathcal{T}$ of in-trees is equal to A, we say that $\mathcal{T}$ *covers* A.

We will study the problem for *covering directed graphs by in-trees* (in short CDGI), and we will present characterizations for a directed graph $D = (V, A, S, f)$ for which there exists a feasible solution of CDGI(D), and we will give a polynomial time algorithm for CDGI(D).

Problem:	CDGI(D)
Input:	a directed graph D;
Output:	a D-feasible set of in-trees which covers the arc set of D, if one exists.

A special class of the problem CDGI(D) in which S consists of a single vertex was considered by Vidyasankar [8] and Frank [4]. They showed the necessary and sufficient condition in terms of linear inequalities that there exists a feasible solution of this problem. However, to the best of our knowledge, an algorithm for CDGI(D) was not presented.

Our Results. We first show that CDGI(D) can be viewed as some type of the connectivity augmentation problem. After this, we will prove that this connectivity augmentation problem can be solved by using an algorithm for the weighted matroid intersection problem in time bounded by a polynomial in $\sum_{i=1}^{d} f(s_i)$ and the size of D. Furthermore, for the case where D is acyclic, we show another characterization for D that there exists a feasible solution of CDGI(D). Moreover, we prove that in this case CDGI(D) can be solved more efficiently than the general case by finding maximum matchings in a series of bipartite graphs.

Outline. The rest of this paper is organized as follows. Section 2 gives necessary definitions and fundamental results. In Section 3, we give an algorithm for the problem CDGI.

2 Preliminaries

Let $D = (V, A, S, f)$ be a connected directed graph which may have multiple arcs. Let $S = \{s_1, \ldots, s_d\}$. Since we can always cover by $|A|$ (D, s_i)-in-trees the arc set of the subgraph of D induced by V_D^i, we consider the problem by using at most $|A|$ (D, s_i)-in-trees. That is, without loss of generality, we assume that $f(s_i) \leq |A|$. For $B \subseteq A$, let $\partial^-(B)$ (resp. $\partial^+(B)$) be a set of tails (resp. heads) of arcs in B. For $e \in A$, we write $\partial^-(e)$ and $\partial^+(e)$ instead of $\partial^-(\{e\})$ and $\partial^+(\{e\})$, respectively. For $W \subseteq V$, we define $\delta_D(W) = \{e \in A \colon \partial^-(e) \in W, \partial^+(e) \notin W\}$. For $v \in V$, we write $\delta_D(v)$ instead of $\delta_D(\{v\})$. For two distinct vertices $u, v \in V$, we denote by $\lambda(u, v; D)$ the local arc-connectivity from u to v in D, i.e., $\lambda(u, v; D) = \min\{|\delta_D(W)| \colon u \in W, v \notin W, W \subseteq V\}$. For $S' \subseteq S$, let $f(S') = \sum_{s_i \in S'} f(s_i)$. For $v \in V$, we denote by $R_D(v)$ a set of vertices in S which are reachable from v in D. For $W \subseteq V$, let $R_D(W) = \bigcup_{v \in W} R_D(v)$.

We call a subgraph T of D *forest* if T has no cycle when we ignore the direction of arcs in T. If a forest T is connected, we call T *tree.* If every arc of an arc set B is parallel to some arc in A, we say that B is *parallel* to A. We denote a directed graph obtained by adding an arc set B to A by $D + B$, i.e., $D + B = (V, A \cup B, S, f)$.

We define D^* as a directed graph obtained from D by adding a new vertex s^* and connecting s_i to s^* with $f(s_i)$ parallel arcs for every $i = 1, \ldots, d$. We denote by A^* the arc set of D^*. We say that D is *(S, f)-admissible* if $|\delta_{D^*}(v)| \leq f(R_D(v))$ holds for any $v \in V$.

2.1 Rooted arc-connectivity augmentation by reinforcing arcs

Given a directed graph $D = (V, A, S, f)$, we call an arc set B with $A \cap B = \emptyset$ which is parallel to A a *D^*-rooted connector* if $\lambda(v, s^*; D^* + B) \geq f(R_D(v))$ holds for every $v \in V$. Notice that since a D^*-rooted connector B is parallel to A, B does not contain an arc which is parallel to an arc entering into s^* in D^*. Then, the problem *rooted arc-connectivity augmentation by reinforcing arcs* (in short RAA-RA) is formally defined as follows.

Problem:	RAA-RA(D^*)
Input:	D^* of a directed graph D;
Output:	a minimum size D^*-rooted connector.

2.2 Matroids on arc sets of directed graphs

In this subsection, we define two matroids $M(D^*)$ and $U(D^*)$ on A^* for a directed graph $D = (V, A, S, f)$, which will be used in the subsequent discussion. We denote by $M = (E, \mathcal{I})$ a matroid on E whose collection of independent sets is $\mathcal{I}$.

For $i = 1, \ldots, d$ and $j = 1, \ldots, f(s_i)$, we define $M_{i,j}(D^*) = (A^*, \mathcal{I}_{i,j}(D^*))$ where $I \subseteq A^*$ belongs to $\mathcal{I}_{i,j}(D^*)$ if and only if both of a tail and a head of every arc in I are contained in $V_D^i \cup \{s^*\}$ and a directed graph $(V_D^i \cup \{s^*\}, I)$ is a forest. $M_{i,j}(D^*)$ is clearly a matroid (i.e. graphic matroid). Moreover, we denote the union of $M_{i,j}(D^*)$ for $i = 1, \ldots, d$ and $j = 1, \ldots, f(s_i)$ by $M(D^*) = (A^*, \mathcal{I}(D^*))$ in which $I \subseteq A^*$ belongs to $\mathcal{I}(D^*)$ if and only if I can be partitioned into $\{I_{i,1}, \ldots, I_{i,f(s_i)} : i = 1, \ldots, d\}$ such that each $I_{i,j}$ belongs to $\mathcal{I}_{i,j}(D^*)$. $M(D^*)$ is also a matroid (see Chapter 12.3 in [7]. This matroid is also called *matroid sum*). When $I \in \mathcal{I}(D^*)$ can be partitioned into $\{I_{i,1}, \ldots, I_{i,f(s_i)} : i = 1, \ldots, d\}$ such that a directed graph $(V_D^i \cup \{s^*\}, I_{i,j})$ is a tree for every $i = 1, \ldots, d$ and $j = 1, \ldots, f(s_i)$, we call I a *complete independent set of $M(D^*)$*.

Next we define another matroid. We define $U(D^*) = (A^*, \mathcal{J}(D^*))$ where $I \subseteq A^*$ belongs to $\mathcal{J}(D^*)$ if and only if I satisfies

$$|\delta_{D^*}(v) \cap I| \leq \begin{cases} f(R_D(v)), & \text{if } v \in V, \\ 0, & \text{if } v = s^*. \end{cases} \tag{1}$$

Since $U(D^*)$ is a direct sum of uniform matroids, $U(D^*)$ is also a matroid (see Exercise 7 of pp.16 and Example 1.2.7 in [7]). We call $I \in \mathcal{J}(D^*)$ a *complete independent set of $U(D)$* when (1) holds with equality.

For two matroids $M(D^*)$ and $U(D^*)$, we call an arc set $I \subseteq A^*$ *D^*-intersection* when $I \in \mathcal{I}(D^*) \cap \mathcal{J}(D^*)$. If a D^*-intersection I is a complete independent set of both $M(D^*)$ and $U(D^*)$, we call I *complete*. When we are given a weight function $w \colon A^* \to \mathbb{R}_+$ where $\mathbb{R}_+$ denotes the set of non-negative reals, we define the weight of $I \subseteq A^*$ (denoted by $w(I)$) by the sum of weights of all arcs I. The *minimum weight complete intersection problem* (in short MWCI) is then defined as follows.

Problem:	MWCI(D^*)
Input:	D^* of a directed graph D and a weight function $w \colon A^* \to \mathbb{R}_+$;
Output:	a minimum weight complete D^*-intersection, if one exists.

Lemma 2.1 *MWCI(D^*) can be solved in $O(M|A^*|^6)$ time where $M = \sum_{v \in V} f(R_D(v))$.*

2.3 Results from [5]

In this section, we introduce results concerning packing of in-trees given by Kamiyama et al. [5] which plays a crucial role in this paper.

Theorem 2.2 ([5]) *Given a directed graph* $D = (V, A, S, f)$*, the following three statements are equivalent: (i) For every* $v \in V$, $\lambda(v, s^*; D^*) \geq f(R_D(v))$ *holds. (ii) There exists a* D*-feasible set of arc-disjoint in-trees. (iii) There exists a complete* D^**-intersection.*

From Theorem 2.2, we obtain the following corollary.

Corollary 2.3 *Given a directed graph* $D = (V, A, S, f)$ *and an arc set* B *with* $A \cap B = \emptyset$ *which is parallel to* A*, the following three statements are equivalent: (i)* B *is a* D^**-rooted connector. (ii) There exists a* $(D + B)$*-feasible set of arc-disjoint in-trees. (iii) There exists a complete* $(D + B)^*$*-intersection.*

Although the following theorem is not explicitly proved in [5], we can easily obtain it from the proof of Theorem 2.2 in [5].

Theorem 2.4 ([5]) *Given a directed graph* $D = (V, A, S, f)$ *which satisfies the condition of Theorem 2.2, we can find a* D*-feasible set of arc-disjoint in-trees in* $O(M^2|A^*|^2)$ *time where* $M = \sum_{v \in V} f(R_D(v))$.

3 An Algorithm for Covering by In-trees

Given a directed graph $D = (V, A, S, f)$, we present in this section an algorithm for CDGI(D). The time complexity of the proposed algorithm is bounded by a polynomial in $f(S)$ and the size of D. We first prove that CDGI(D) can be reduced to RAA-RA(D^*). After this, we show that RAA-RA(D^*) can be reduced to the problem MWCI.

3.1 Reduction from CDGI to RAA-RA

If $D = (V, A, S, f)$ is not (S, f)-admissible, i.e., $|\delta_{D^*}(v)| > f(R_D(v))$ for some $v \in V$, there exists no feasible solution of CDGI(D) since there can not be a D-feasible set of in-trees that covers $\delta_D(v)$ from the definition of a D-feasible set of in-trees. Thus, we assume in the subsequent discussion that D is (S, f)-admissible. For an (S, f)-admissible directed graph $D = (V, A, S, f)$, we define $\mathsf{opt}_D = \sum_{v \in V} f(R_D(v)) - (|A| + f(S))$. It is not difficult to see that the size of a D^*-rooted connector is at least opt_D. From Corollary 2.3, we obtain the following lemma.

Lemma 3.1 *Given an* (S, f)*-admissible directed graph* $D = (V, A, S, f)$*, there exists a feasible solution of CDGI(*D*) if and only if there exists a* D^**-rooted connector whose size is* opt_D.

Although the details are omitted, from the proof of Lemma 3.1, if we can find a D^*-rooted connector B with $|B| = \mathsf{opt}_D$, we can compute a D-feasible set of in-trees $T_{i,j}$ for $i = 1, \ldots, d$ and $j = 1, \ldots, f(s_i)$ which covers A by using the following procedure Replace from a $(D + B)$-feasible set of arc-disjoint in-trees $T'_{i,j}$ for $i = 1, \ldots, d$ and $j = 1, \ldots, f(s_i)$.

Replace: For $i = 1, \ldots, d$ and $j = 1, \ldots, f(s_i)$, set $T_{i,j}$ to be a directed graph obtained by replacing every arc $e \in B$ which is contained in $T'_{i,j}$ by an arc in A which is parallel to e.

Furthermore, we can construct a $(D+B)$-feasible set of arc-disjoint in-trees by using the algorithm of Theorem 2.4. Since the optimal value of RAA-RA(D^*) is at least opt_D, we can test if there exists a D^*-rooted connector whose size is equal to opt_D by solving RAA-RA(D^*). Assuming that we can solve RAA-RA(D^*), our algorithm for finding a D-feasible set of in-trees which covers A called Algorithm CR can be illustrated as Algorithm 1 below.

```
Algorithm 1 Algorithm CR
Input: a directed graph $D = (V, A, S, f)$
Output: a $D$-feasible set of in-trees covering $A$, if one exists
 1: if $D$ is not $(S, f)$-admissible then
 2:    Halt (there exists no $D$-feasible set of in-trees covering $A$)
 3: end if
 4: Find an optimal solution $B$ of RAA-RA$(D^*)$
 5: if $|B| > \mathsf{opt}_D$ then
 6:    Halt (there exists no $D$-feasible set of in-trees covering $A$)
 7: else
 8:    Construct a $(D+B)$-feasible set $\mathcal{T}'$ of arc-disjoint in-trees
 9:    Construct a set $\mathcal{T}$ of in-trees from $\mathcal{T}'$ by using the procedure Replace
10:    return $\mathcal{T}$
11: end if
```

From Theorem 2.4 and Lemma 3.1, we obtain the following lemma.

Lemma 3.2 *Given a directed graph $D = (V, A, f, S)$, Algorithm* CR *correctly finds a D-feasible set of in-trees which covers A in $O(\gamma_1 + |V||A| + M^4)$ time if one exists where γ_1 is the time required to solve RAA-RA(D^*) and $M = \sum_{v \in V} f(R_D(v))$.*

3.2 Reduction from RAA-RA to MWCI

From the algorithm CR in Section 3.1, in order to present an algorithm for CDGI(D), what remains is to show how we solve RAA-RA(D^*). In this section, we will prove that we can test whether there exists a D^*-rooted connector whose size is equal to opt_D (i.e., Steps 4 and 5 in the algorithm CR) by reducing it to the problem MWCI. Our proof is based on the algorithm of [3] for RAA-RA(D^*) for $D = (V, A, S, f)$ with $|S| = 1$. We extend the idea of [3] to the general case by using Theorem 2.2. We define a directed graph D_+ obtained from an (S, f)-admissible directed graph $D = (V, A, S, f)$ by adding opt_D parallel arcs to every $e \in A$. Then, we will compute a D^*-rooted connector whose size is equal to opt_D by using an algorithm for MWCI(D_+^*) as described below. Since the number of arcs in a D^*-rooted connector whose size is equal to opt_D which are parallel to one arc in A is at most opt_D, it is enough to add opt_D parallel arcs to each arc of A in D_+ in order to find a D^*-rooted connector whose size is equal to opt_D.

We denote by A_+^* the arc sets of D_+^*. We define a weight function $w\colon A_+^* \to \mathbb{R}_+$ by

$$w(e) = \begin{cases} 0, & \text{if } e \in A^*, \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

We can prove the following lemma by using Corollary 2.3.

Lemma 3.3 *Given an (S,f)-admissible directed graph $D=(V,A,S,f)$ and a weight function $w\colon A^*_+ \to \mathbb{R}_+$ defined by (2), there exists a D^*-rooted connector whose size is equal to* opt_D *if and only if there exists a complete D^*_+-intersection whose weight is equal to* opt_D.

Although the details are omitted, from the proof of Lemma 3.3, if we can find a complete D^*_+-intersection I with $w(I)=\mathsf{opt}_D$, we can find a D^*-rooted connector B with $|B|=\mathsf{opt}_D$ by setting $B=I\setminus A^*$. Furthermore, we can obtain a complete D^*_+-intersection whose weight is equal to opt_D if one exists by using the algorithm for MWCI(D^*_+) since it is not difficult to see that the optimal value of MWCI(D^*_+) is at least opt_D. The formal description of the algorithm called Algorithm RM for finding a D^*-rooted connector whose size is equal to opt_D is illustrated in Algorithm 2.

Algorithm 2 Algorithm RM

```
Input: $D^*$ of an $(S,f)$-admissible directed graph $D=(V,A,S,f)$
Output: a $D^*$-rooted connector whose size is equal to $\mathsf{opt}_D$, if one exists
Find an optimal solution $I$ for MWCI($D^*_+$) with a weight function $w$ defined by (2)
if there exists no solution of MWCI($D^*_+$) or $w(I) > \mathsf{opt}_D$ then
    Halt (There exists no $D^*$-rooted connector whose size is equal to $\mathsf{opt}_D$)
end if
return $I \setminus A^*$
```

The lemma immediately follows from Lemma 3.3.

Lemma 3.4 *Given D^* of an (S,f)-admissible directed graph $D=(V,A,f,S)$, Algorithm* RM *correctly finds a D^*-rooted connector whose size is equal to* opt_D *in $O(\gamma_2+M|A|)$ time if one exists where γ_2 is the time required to solve MWCI(D^*_+) and $M=\sum_{v\in V} f(R_D(v))$.*

3.3 Algorithm for CDGI

We are ready to explain the formal description of our algorithm called Algorithm Covering for CDGI(D). Algorithm Covering is the same as Algorithm CR such that Steps 4, 5 and 6 are replaced by Algorithm RM. Then, the following theorem follows from Lemmas 2.1, 3.2 and 3.4.

Theorem 3.5 *Given a directed graph $D=(V,A,S,f)$, Algorithm* Covering *correctly finds a D-feasible set of in-trees which covers A in $O(M^7|A|^6)$ time if one exits where $M=\sum_{v\in V} f(R_D(v))$.*

References

[1] E. M. Arkin, R. Hassin, and A. Levin. Approximations for minimum and min-max vehicle routing problems. *J. Algorithms*, 59(1):1–18, 2006.

[2] G. Even, N. Garg, J. Könemann, R. Ravi, and A. Sinha. Min-max tree covers of graphs. *Oper. Res. Lett.*, 32(4):309–315, 2004.

[3] A. Frank. Rooted k-connections in digraphs. *Discrete Applied Mathematics.* (to appear).

[4] A. Frank. Covering branchings. *Acta Scientiarum Mathematicarum [Szeged]*, 41:77–81, 1979.

[5] N. Kamiyama, N. Katoh, and A. Takizawa. Arc-disjoint in-trees in directed graphs. *Combinatorica.* to appear. Preliminary version in *Proc. the nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA2008)*, pp.518-526.

[6] H. Nagamochi and K. Okada. Approximating the minmax rooted-tree cover in a tree. *Inf. Process. Lett.*, 104(5):173–178, 2007.

[7] J. G. Oxley. *Matroid theory*. Oxford University Press, 1992.

[8] K. Vidyasankar. Covering the edge set of a directed graph with trees. *Discrete Mathematics*, 24:79–85, 1978.