

## APPROXIMATION ALGORITHMS FOR METRIC TREE COVER AND GENERALIZED TOUR AND TREE COVERS

VIET HUNG NGUYEN<sup>1</sup>

**Abstract.** Given a weighted undirected graph  $G = (V, E)$ , a tree (respectively tour) cover of an edge-weighted graph is a set of edges which forms a tree (resp. closed walk) and covers every other edge in the graph. The tree (resp. tour) cover problem is of finding a minimum weight tree (resp. tour) cover of  $G$ . Arkin, Halldórsson and Hassin (1993) give approximation algorithms with factors respectively 3.5 and 5.5. Later Könemann, Konjevod, Parekh, and Sinha (2003) study the linear programming relaxations and improve both factors to 3. We describe in the first part of the paper a 2-approximation algorithm for the metric case of tree cover. In the second part, we will consider a generalized version of tree (resp. tour) covers problem which is to find a minimum tree (resp. tours) which covers a subset  $D \subseteq E$  of  $G$ . We show that the algorithms of Könemann *et al.* can be adapted for the generalized tree and tours covers problem with the same factors.

**Keywords.** Approximation algorithms, graph algorithms, network design.

**Mathematics Subject Classification.** 90C27, 90C59.

### INTRODUCTION

Let  $G = (V, E)$  be an undirected graph with a (nonnegative) cost function  $c : E \Rightarrow \mathbb{Q}_+$  defined on the edges and let  $D \subseteq E$  be an edge subset. A *tree* (resp. *tour*) covers  $D$  is a subgraph  $T = (U, F)$  such that

---

Received June 17, 2006. Accepted December 12, 2006.

<sup>1</sup> LIP6 - Université Pierre et Marie Curie - Paris 6, 4 place Jussieu, 75252 Paris Cedex, France; [hung.nguyen@lip6.fr](mailto:hung.nguyen@lip6.fr)

© EDP Sciences, ROADEF, SMAI 2007

- (1) for every  $e \in D$ ,  $F$  contains an edge  $f$  adjacent to  $e$ , *i.e.*  $F \cap N(e) \neq \emptyset$  where  $N(e)$  is the set of the edges adjacent to  $e$ .
- (2)  $T$  is a tree (resp. closed walk).

The generalized tree (resp. tour) cover problem consists in finding a tree (resp. tour) cover of minimum total weight:

$$\min \sum_{e \in F} c_e,$$

over subgraphs  $T = (U, F)$  which form a tree (resp. tour) cover of  $D$  in  $G$ .

When  $D = E$ , these problems are simply called tree and tour cover problems and were introduced by Arkin, Haldórsson and Hassin [1]. In this case, the two problems are shown in [1] to be *NP*-hard. The tree cover problem is even hard when all edge weights are equal, since it is asking for a minimum connected vertex cover (see [3], p. 190). The tour cover problem can be reduced to the Traveling Salesman Problem by creating in the instance of the TSP an edge between each node and adding a new artificial node of degree 1. The motivation for the study of these problems comes from the close relation of these problems to vertex cover, watchman route and traveling purchaser problems. Arkin et al. provide fast combinatorial algorithms for the weighted version of this problem achieving approximation factors 5.5. Later Könemann, Konjevod, Parekh, and Sinha [6] improved the approximation factors by considering integer formulations and their linear relaxations. They obtain a factor 3 approximation algorithm for both tree and tour covers. Note that in the case of tour cover, the tour is not constrained to be a simple cycle, thus we can replace every edge  $uv$  in a tour by the shortest path between  $u$  and  $v$  if this later is not  $uv$  to obtain another tour with smaller cost. Therefore one can assume without loss of generality that the edge costs satisfy triangle inequality. But for tree cover, we can not replace an edge of a tree by the shortest path between it's endnodes to obtain another tree. Thus in general one can not assume that the edge costs satisfy triangle inequality. Therefore one can hope to obtain a better approximation factor for metric tree cover problem. In fact, in Section 1, we describe a 2-approximation algorithm for this case.

Now let us consider the case when  $D \subset E$ . In this case, the two problems remain *NP*-hard. They can be reduced respectively to the Steiner tree and the TSP problem by the same way as for the case when  $D = E$ , *i.e.* by adding an edge between each node and an artificial node of degree 1. One can set  $D$  as the set consisting of the new edges. As Arkin et al. have remarked in their paper, their algorithms don't work when  $D \subset E$ . Indeed, these algorithms find a collection of subtours that covers the edges of  $G$  in the case of tour cover and a forest cover in the case of tree cover and then connect them by a Traveling salesman tour or a Steiner tree. But this Traveling salesman tour and this Steiner tree are in general not an approximation of respectively generalized tour and tree cover since the subgraphs induced by subtours and forest may not contain any edge belonging to  $D$ . They describe however a modification of their algorithm which works only for the unweighted generalized tree cover problem and achieves a factor 2.46. In their

paper, Könemann et al. do not discuss the generalized problems. In Section 2, we show that the algorithms given by Könemann et al. do not work for the generalized tree and tour covers problems and give an adaptation of their algorithms which gives the same factor 3 for both generalized problems.

Let us introduce the notation that will be used in the paper. Let  $G = (V, E)$  be a graph with vertex set  $V$  and edge set  $E$ . If  $x \in \mathbb{R}^{|E|}$  is a vector indexed by the edge set  $E$  and  $F \subseteq E$  is a subset of edges, we use  $x(F)$  to denote the sum of values of  $x$  on the edges in the set  $F$ ,  $x(F) = \sum_{e \in F} x_e$ . Similarly, for a vector  $y \in \mathbb{R}^{|V|}$  indexed by the vertices of a graph  $G = (V, E)$  and  $S \subseteq V$  is a subset of vertices,  $y(S)$  denotes the sum of values of  $y$  on the vertices in the set  $S$ . For a subset of vertices  $S \subseteq V$ , we write  $\delta_G(S)$  for the set of edges in the graph  $G$  with exactly one endnode inside  $S$  and  $E(S)$  for the set of edges with both endnodes inside  $S$ . If  $G$  is directed let  $\delta^+(G)$  (resp.  $\delta^-(G)$ ) be the the set of edges having only the tail (resp. head) in  $S$ . For two subset  $U, W \subset V$  such that  $U \cap W = \emptyset$ , let  $(U : W)$  be the set of the edges having one endnode in  $U$  and the other in  $W$ . If  $G$  is directed, let  $(U : W)$  (resp.  $(W : U)$ ) be the set of the edges having the tail (resp. head) in  $U$  and the head (resp. tail) in  $W$ .

When we work on more than one graph, we precise the graph in the index of the notation, *e.g.*  $\delta_G(S)$  will denote  $\delta(S)$  in the graph  $G$ .

### 1. TREE COVER WITH METRIC COSTS

In this section, we consider the tree cover problem when the costs  $c_{ij}$  satisfy the triangular inequality. That is

$$c_{ij} \leq c_{ik} + c_{kj} \text{ for all edge triplets } ij, ik \text{ and } kj \text{ in } E.$$

We use an integer formulation similar to the one described in [6]. First, we transform the original graph  $G$  into a directed graph  $\vec{G} = (V, \vec{E})$  by replacing every undirected edge  $uv$  by a pair of directed edges  $(u, v)$ ,  $(v, u)$  each having the same weight as the original undirected edge. The formulation is as follows: for a fixed root  $r$ , define  $\mathcal{F}$  to be the set of all subsets  $S$  of  $V \setminus \{r\}$  such that  $S$  induces at least one edge of  $\vec{E}$ ,

$$\mathcal{F} = \{S \subseteq V \setminus \{r\} \mid \vec{E}(S) \neq \emptyset\}.$$

Let  $C$  be the edge set of a tree cover of  $G$  containing  $r$  and  $\vec{C}$  be the corresponding branching with the root  $r$ . Now for every  $S \in \mathcal{F}$ ,  $\vec{C}$  must contain at least one edge entering  $S$ . Hence we have the following IP formulation.

$$\begin{aligned} \min \quad & \sum_{e \in \vec{E}} c_e x_e \\ & \sum_{e \in \delta^-(S)} x_e \geq 1 \text{ for all } S \in \mathcal{F} \\ & x \in \{0, 1\}^{|\vec{E}|}. \end{aligned}$$

A trivial case for which this formulation has no constraint is when  $G$  is a star but in this case the optimal solution is trivially the central node with cost 0.

Replacing the integrality constraints by

$$x \geq 0,$$

we obtain the linear programming relaxation. We use the  $\text{TrC}(\vec{G})$  to denote the convex hull of all vectors  $x$  satisfying the constraints above (with integrality constraints replaced by  $x \geq 0$ ).

### 1.1. $r$ -ARBORESCENCE POLYHEDRON

Let us consider  $\vec{G} = (V, \vec{E})$ , an  $r$ -arborescence of  $\vec{G}$  is a spanning tree (when  $\vec{G}$  is viewed as an undirected graph) directed away from  $r$ . It is well-known that when the costs are non-negative, the minimum cost  $r$ -arborescence can be found by solving the following linear program:

$$\begin{aligned} \min \quad & \sum_{e \in \vec{E}} c_e x_e \\ & \sum_{e \in \delta^-(S)} x_e \geq 1 \text{ for all } S \subset V \setminus \{r\} \\ & x \geq 0. \end{aligned} \tag{1}$$

Let us call  $\text{RA}(G)$  the polyhedron defined by (1) and the non-negativity constraint. Polynomial time algorithms that solve the minimum cost  $r$ -arborescence problem was given in [2, 5].

### 1.2. ALGORITHM

We are now ready to state our algorithm for metric tree cover.

(1) For every vertex  $r \in V$ , let  $x_r^*$  be the vector minimizing  $cx$  over  $\text{TrC}(\vec{G})$  with root  $r$ .

(2) Let  $U \leftarrow \{v \in V \mid x_r^*(\delta^-(\{v\})) \geq \frac{1}{2}\} \cup \{r\}$ .

(3) Let  $G_U$  be the subgraph of  $G$  induced by  $U$  with the same cost  $c_e$  for each edge  $e$  in  $G_U$ . Find the minimum cost  $r$ -arborescence of  $G_U$  and call this  $T_r$ .

(4) Pick the cheapest such  $T_r$ .

Note that the linear program in step (1) can be solved in polynomial time by using the ellipsoid method with a min-cut computation as separation oracle. The algorithm initially yields a branching in the bidirected graph. We map this in the obvious way to a set of edges in the original undirected graph.

The algorithm outputs a tree which spans  $U$ . We can see that  $U$  is vertex cover of  $G$ , since for any edge  $e = uv \in E$ ,  $x_r^*(\delta^-(\{u, v\})) \geq 1$ , at least  $u$  or  $v$  should belong to  $U$ . Therefore, the algorithm outputs a tree cover of  $G$ .

1.3. PERFORMANCE GUARANTEE

**Theorem 1.** *Let  $x_r^*$  be a vector minimizing  $cx$  over  $TrC(\vec{G})$  and  $U = \{v \in V \mid x_r^*(\delta^-(\{v\})) \geq \frac{1}{2}\}$ . Then*

$$\min\{cy \mid y \in RA(G_U)\} \leq 2 * cx^* = 2 \min\{cx \mid x \in TrC(G)\}.$$

*Proof.* Let  $W = V \setminus U$ , note that  $W$  is a stable set in  $G$ . We build a vector  $z^* \in \mathbb{R}^{\vec{E}}$  from  $x^*$  as follows:

- (1)  $z_e^* \leftarrow x_e^*$  for all  $e \in \vec{E}$ .
- (2) While there exists  $w \in W$  and two distinct vertices  $u, v \in U$  such that  $z_{(u,w)}^* > 0$  and  $z_{(w,v)}^* > 0$  then set

$$z^*(u, v) \leftarrow z_{(u,v)}^* + \min\{z_{(u,w)}^*, z_{(w,v)}^*\},$$

$$z_{(u,w)}^* \leftarrow z_{(u,w)}^* - \min\{z_{(u,w)}^*, z_{(w,v)}^*\},$$

$$z_{(w,v)}^* \leftarrow z_{(w,v)}^* - \min\{z_{(u,w)}^*, z_{(w,v)}^*\}.$$

Otherwise, STOP.

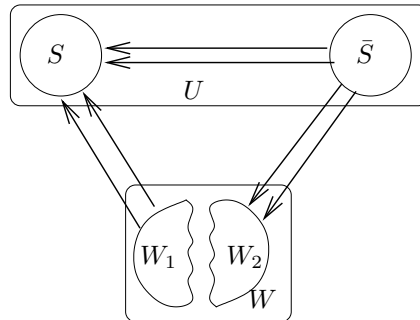
As  $c_{(u,w)} + c_{(w,v)} \geq c_{(u,v)}$ , we have  $cz^* \leq cx^*$ .

Let  $y^* = 2z^*$ , in order to prove Theorem 1, it is sufficient to show that  $y^* \in RA(G_U)$ . Assume that it is not true then there exists a set  $S \subset U \setminus \{r\}$  such that  $y^*(\delta_{G_U}^-(S)) < 1$  in  $G_U$ . Let  $\bar{S} = U \setminus S$ . By the construction of  $z^*$ , there is no vertex  $w \in W$  such that  $y^*(\bar{S} : \{w\}) > 0$  and  $y^*(\{w\} : S) > 0$ . Thus, we can partition  $W$  into two subsets  $\emptyset \subseteq W_1 \subseteq W$  and  $\emptyset \subseteq W_2 \subseteq W$  as follows: For every vertex  $w \in W$ :

- if  $y^*(\bar{S} : \{w\}) = 0$  and  $y^*(\{w\} : S) > 0$  then let us put  $w$  in  $W_1$ ;
- if  $y^*(\bar{S} : \{w\}) > 0$  and  $y^*(\{w\} : S) = 0$  then let us put  $w$  in  $W_2$ ;
- if  $y^*(\bar{S} : \{w\}) = 0$  and  $y^*(\{w\} : S) = 0$  then let us put  $w$  in  $W_2$ .

We can see that this partition  $W_1, W_2$  of  $W$  satisfies the following

- $y^*(\bar{S} : W_1) = 0$  and  $y^*(W_2 : S) = 0$ ;
- either  $W_1 = \emptyset$  or  $y^*(W_1 : S) > 0$ .



Let us consider the case when  $W_1 = \emptyset$ , i.e.  $W_2 = W$ , then as  $y^*(W_2 : S) = 0$ , i.e.  $z^*(W_2 : S) = 0$ , by the construction of  $z^*$ , we have

$$z^*(\delta_{G_U}^-(S)) = x_{\vec{G}}^*(\vec{S} : S) + x_{\vec{G}}^*(W : S) = x_{\vec{G}}^*(\delta^-(S)).$$

As  $S \subset U$ ,  $x_{\vec{G}}^*(\delta^-(S)) \geq \frac{1}{2}$ , thus we derive that  $y^*(\delta_{G_U}^-(S)) \geq 1$ . Contradiction.

Let us now consider the remaining case when  $y^*(W_1 : S) > 0$ , we can see that  $\vec{E}(S \cup W_1) \neq \emptyset$ . Thus  $x_{\vec{G}}^*(\delta^-(S \cup W_1)) \geq 1$ . But as  $y^*(\vec{S} : W_1) = 0$  and  $y^*(W_2 : S) = 0$  then

$$z^*(\delta_{G_U}^-(S)) = x_{\vec{G}}^*(\delta^-(S \cup W_1)) \geq 1.$$

We derive that  $y^*(\delta_{G_U}^-(S)) \geq 2$ . Contradiction.  $\square$

**Corollary 1.** *When the costs satisfy triangle inequality, the algorithm in 1.2. outputs a tree cover of cost no more than 2 time the cost of the minimum tree cover.*

## 2. GENERALIZED TOUR AND TREE COVER PROBLEMS

We consider firstly the generalized tour cover problem. As we argue in the introduction section, for the (generalized) tour cover problem, because a tour is a closed walk which can repeat nodes, one can consider that the cost vector  $c$  satisfies triangle inequality. If the cost vector  $c$  does not satisfy triangle inequality, for each edge  $e = uv$ , we can replace  $c_e$  by the value the shortest elementary chain evaluated with the original cost vector  $c$  between  $u$  and  $v$  in  $G$ . Then the new cost vector satisfies triangle inequality and this does not change the optimal solutions of the problem.

Let  $\mathcal{F}$  denote the set of all subsets  $S$  of  $V$  such that both  $S$  and  $V \setminus S$  induce at least one edge of  $E$ ,

$$\mathcal{F} = \{S \subseteq V \mid E(S) \neq \emptyset, E(V \setminus S) \neq \emptyset\}.$$

Könemann et al. observe that if  $U$  is a set of edges that forms a tour cover of  $G$ , then at least 2 edges of  $U$  cross  $S$ , for every  $S \in \mathcal{F}$ . This observation motivates an integer formulation of tour cover. For every edge  $e \in E$ , let the integer variable  $x_e$  indicate the number of copies of  $e$  included in the tour cover. One minimizes the total weight of edges included, under the condition that every cut in  $\mathcal{F}$  be crossed at least twice. In order to ensure that the solution is a tour, each vertex is needed to have even degree; however as in [6] we can drop these constraints and consider

the following relaxation.

$$\begin{aligned} & \min \sum_{e \in E} c_e x_e \\ & \sum_{e \in \delta(S)} x_e \geq 2 \quad \text{for all } S \in \mathcal{F} \\ & x \in \{0, 1, 2\}^{|E|}. \end{aligned} \tag{2}$$

For the generalized tour cover problem, we can keep the same formulation applying to the following re-definition of the set  $\mathcal{F}$ :

$$\mathcal{F} = \{S \subseteq V \mid E(S) \cap D \neq \emptyset, E(V \setminus S) \cap D \neq \emptyset\}.$$

Replacing the integrality constraints by

$$0 \leq x \leq 2,$$

we obtain the linear programming relaxation. We call  $\text{ToC}(G)$  the convex hull of all feasible solutions of this linear relaxation. The separation problem of the constraints (2) can be solved in polynomial time by computing minimal capacity cut in  $G$  that separates each pair of edges  $e_1, e_2 \in D$ . Thus optimizing a linear function over  $\text{ToC}(G)$  can be done in polynomial time by the ellipsoid method.

Let  $x^*$  be the vector minimizing  $cx$  over  $\text{ToC}(G)$ . Let us consider the case when  $D = E$ , *i.e.* the tour cover problem, Könemann et al. solve the linear programming relaxation and determine from the  $x^*$  a vertex cover  $U$ . Their final solution is a TSP over  $U$ . Precisely,  $U$  is defined as

$$U = \{v \in V \mid x^*(\delta(\{v\})) \geq 1\}.$$

We can see that for each edge  $e = uv$ , at least one of the vertices  $u$  and  $v$  belongs to  $U$  because (2) is satisfied by  $x^*$ . Thus  $U$  is a vertex cover, in addition,  $U$  has the *cut property* that if we let  $y = 2x^*$  then  $y$  is feasible for

$$\begin{aligned} A = \{x \geq 0 \mid & x(\delta(v)) \geq 0 \quad \forall v \in V \setminus U \\ & x(\delta(u)) \geq 2 \quad \forall u \in U \\ & x(\delta(S)) \geq 2 \quad \forall S \subset V, S \cap U \neq \emptyset, U \setminus S \neq \emptyset, \emptyset \neq S \neq V \\ & x(\delta(S)) \geq 0 \quad \forall S \subset V \setminus U, S \neq \emptyset\} \end{aligned}$$

by parsimonious property [4],  $\min\{cx \mid x \in A\} = \min\{cx \mid x \in B^0\}$  with

$$\begin{aligned} B^0 = \{x \geq 0 \mid & x(\delta(v)) = 0 \quad \forall v \in V \setminus U \\ & x(\delta(u)) = 2 \quad \forall u \in U \\ & x(\delta(S)) \geq 2 \quad \forall S \subset V, S \cap U \neq \emptyset, U \setminus S \neq \emptyset, \emptyset \neq S \neq V \\ & x(\delta(S)) \geq 0 \quad \forall S \subset V \setminus U, S \neq \emptyset\} \end{aligned}$$

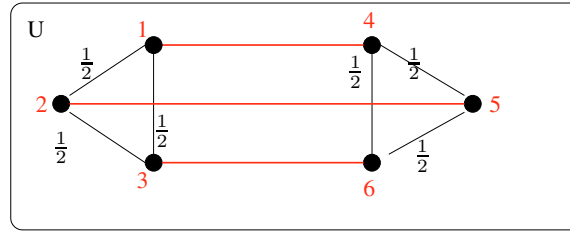
Let  $F$  be the complete graph over  $U$  with the shortest chain cost  $c_e$  on every edge  $e$ . Let

$$B = \{x \geq 0 \mid \begin{aligned} x(\delta(v)) &= 0 \ \forall v \in V \setminus U \\ x(\delta(u)) &= 2 \ \forall u \in U \\ x(\delta(S)) &\geq 2 \ \forall S \subset U, \emptyset \neq S \neq U \}. \end{aligned}$$

$B$  denotes the subtour polytope over  $F$ ,  $ST(F)$  and Könemann et al. show that  $B^0 = B$ . This implies  $\min\{cx \mid x \in B\} = \min\{cx \mid x \in A\}$ . Wolsey [9] and Shmoys and Williamson [7] have shown that the solution given by Christophides's heuristic over the graph  $F$  is not worse than  $3/2$  of the optimal solution over  $B$  ( $ST(F)$ ). Thus Könemann et al. apply the Christophides's heuristic on the graph  $F$  to get a solution of the tour cover problem which is not worst than  $(3/2)y = (3/2) * 2x^* = 3x^*$ . We show in the following remark that in the case when  $D \subset E$ , the set  $U$  chosen as in the Köneman et al.'s algorithm may not have the cut property.

**Remark 1.** For generalized tour cover, the set  $U$  do not have necessarily the cut property.

*Proof.* There can be a subset  $S \subset U$  such that  $E(S) \neq \emptyset$  and  $E(S) \cap D = \emptyset$ . As the integer formulation does not impose that  $x(\delta(S)) \geq 2$ , one can not guarantee that  $x^*(\delta(S)) \geq 2$ . As in the figure below with  $U = \{1, 2, 3, 4, 5, 6\}$ ,



— : the edges  $e$  such that  $x_e^* \neq 0$  ( $= \frac{1}{2}$ ).  
 — : the edges in  $D$ .

if we take  $S = \{1, 2, 3\} \subset U$  then  $x^*(\delta(S)) = 0 < 2$ . □

In order to obtain a set  $U$  which is a vertex cover of  $D$  and has the cut property, we propose in the following a procedure of refinement of the set  $U$ :

**Procedure 1. Refinement of the set  $U$  for generalized tour cover (resp. tree cover).**

**Input:**  $G$  (resp.  $\vec{G}$ ),  $U$  and  $x^*$ .

**Output:** A new set  $U$ .

- (1) In the graph  $G$  weighted by  $x^*$ , find the minimum capacity cut (resp. directed cut) that separates any two vertices of  $U$ . If the value of the cut is greater than 1 (resp.  $\frac{1}{2}$ ) then STOP. Otherwise let  $\delta(S)$  (resp.  $\delta^-(S)$ ) be this cut with  $S \cap U \neq \emptyset$  and  $(V \setminus S) \cap U \neq \emptyset$  and go to 2.



(2) Let  $U \leftarrow U \setminus (S \cap U)$  and go to 1.

**Corollary 2.** *The set  $U$  resulting from Procedure 1 has the cut property, i.e. let  $y = 2x^*$ ,  $y$  is feasible for*

$$A = \{x \geq 0 \mid \begin{array}{l} x(\delta(v)) \geq 0 \ \forall v \in V \setminus U \\ x(\delta(u)) \geq 2 \ \forall u \in U \\ x(\delta(S)) \geq 2 \ \forall S \subset V, S \cap U \neq \emptyset, U \setminus S \neq \emptyset, \emptyset \neq S \neq V \\ x(\delta(S)) \geq 0 \ \forall S \subset V \setminus U, S \neq \emptyset \}. \end{array}$$

*Proof.* It is straightforward from the construction of  $U$  in Procedure 1.  $\square$

**Lemma 1.** *The set  $U$  resulting from Procedure 1 is a vertex cover of  $D$ .*

*Proof.* Assume that at some iteration, let  $U' = \{u' \mid x^*(\delta(u')) \geq 1 \text{ and } u' \notin U\}$ , i.e.  $U'$  contains the vertices excluded from  $U$  so far and we find a cut  $\delta(S)$  whose capacity is strictly less than 1 such that  $S \cap U \neq \emptyset$  and  $(V \setminus S) \cap U \neq \emptyset$ . If the set  $U \setminus (S \cap U)$  is not a cover of  $D$ , i.e. there exists a vertex  $t \in V \setminus (S \cup U)$  and a vertex  $s \in S \cap U$  such that the edge  $st \in D$ . There are 2 possible cases:

- $t \notin U'$ , i.e.  $x^*(\delta(t)) < 1$ . As  $x^*(\delta(S)) < 1$ , we deduce that  $x^*(\delta(S \cup \{t\})) < 2$ . But as  $(S \cup \{t\}) \in \mathcal{F}$  and  $x^*$  is a solution of (2) then  $x^*(\delta(S \cup \{t\})) \geq 2$ . Contradiction.
- $t \in U'$ . Let  $S_t \subseteq U'$  be the subset containing  $t$  that was excluded from  $U$  in some previous iteration, we have  $x^*(\delta(S_t)) < 1$ . As  $x^*(\delta(S)) < 1$ , we deduce that  $x^*(\delta(S \cup S_t)) < 2$ . But since  $(S \cup S_t) \in \mathcal{F}$  and  $x^*$  is a solution of ToC( $G$ ), we have  $x^*(\delta(S \cup S_t)) \geq 2$ . Contradiction.

We deduce therefore that  $U \setminus (S \cap U)$  is a cover of  $D$ .  $\square$

**Theorem 2.** *The generalized tour cover problem can be approximated by a factor 3.*

*Proof.* We can directly apply the algorithms of Könemann *et al.* to the set  $U$  resulting from Procedure 1.  $\square$

We can apply the same technique for the generalized tree cover problem. Let  $x_r^*$  be an optimal solution obtained by solving the linear relaxation associated to a specific root node  $r$  (see Sect. 1). Könemann *et al.* define  $U = \{v \in V \mid x_r^*(\delta^-(\{v\})) \geq 1/2\}$ . They show that  $U$  is a vertex cover and has the cut property, i.e. let  $y = 2x_r^*$  then  $y$  is feasible for

$$QBST(\vec{G}) = \{x \in \mathbb{Q}_+^{\vec{E}} \mid x(\delta^-(S)) \geq 1 \text{ for all } S \subset V \setminus \{r\}, S \cap U \neq \emptyset\},$$

which defines the linear relaxation for the bidirected Steiner tree problem with  $U$  as the terminal node set. When  $V \setminus U$  is a stable set, finding the minimum weight bidirected Steiner tree in  $\vec{G}$  with the terminal set  $U$  is a special case called minimum weight quasi-bipartite bidirected Steiner tree problem. In this case,

Rajagopalan and Varizani [8] have designed an algorithm giving a Steiner tree whose value is not worse than  $\frac{3}{2}$  the value of the optimal solution of the above linear relaxation. Könemann *et al.* apply the Rajagopalan-Varizani's algorithm on  $\vec{G}$  to obtain a 3-approximation algorithm for the tree cover problem. Applying this to the generalized tree cover problem, we can obtain a set  $U$  for which  $y = 2x_r^*$  does not belong to  $QBST(\vec{G})$ , *i.e.* there can exist some subset  $S$  such that  $S \subset V \setminus \{r\}$ ,  $S \cap U \neq \emptyset$ ,  $D \cap E(S) = \emptyset$  and  $x^*(\delta^-(S)) < \frac{1}{2}$ . Applying the Procedure 1 in the similar way (with searching value for the min-cut is now  $\frac{1}{2}$ ), we can obtain a new set  $U$  that always covers  $D$  and has the cut property. The fact that may be the  $V \setminus U$  now is not anymore a stable set is not troublesome. In this case, we define a vector  $y' \in \vec{E}$  as follows:

$$y'_e = \begin{cases} 0 & \text{if } e \in \vec{E}(V \setminus U) \\ y_e & \text{otherwise.} \end{cases}$$

Since the edge costs are non-negative, we have  $cy' \leq cy$ . In addition, we can see that  $y'$  is feasible for  $QBST(\vec{G}_1)$  where  $\vec{G}_1$  is a the subgraph of  $\vec{G}$  which contains all nodes in  $V$  and all arcs in  $\vec{E}$  except those in  $\vec{E}(V \setminus U)$ . Thus we can consider the minimum weight quasi-bipartite bidirected Steiner tree problem on  $\vec{G}_1$ . Solving this problem will give us a tree cover of  $D$  on  $G$  of weight at most 3 times the weight of the optimal tree.

**Corollary 3.** *We can approximate the generalized tree cover with a factor 3.*

### 3. OPEN PROBLEMS AND DISCUSSION

It seems hard to find a pure combinatorial approximation algorithm with constant factor for generalized tour and tree cover. As we argue in the Introduction section, the principle of the algorithms of Arkin *et al.* can not be applied. Perhaps, one should design an algorithm based on a completely new idea.

It has been remarked in [6] that the dual of the linear relaxation of above integer formulations fits into combinatorial packing framework that enables us to avoid ellipsoid method and to use fast packing combinatorial algorithm. But then we have to pay a factor of  $(1 + \epsilon)$ -factor in the approximation guarantee.

*Acknowledgements.* The author would like to thank the anonymous referee for many helpful comments.

### REFERENCES

- [1] E.M. Arkin, M.M. Halldórsson and R. Hassin, Approximating the tree and tour covers of a graph. *Inf. Process. Lett.* **47** (1993) 275–282.
- [2] J. Edmonds, Optimum branchings. *J. Res. Nat. Bur. Stand. B* **71** (1965).

- [3] M.R. Garey and D.S. Johnson, *Computer and Intractability: A Guide to the theory of the NP-Completeness*, Freeman (1978).
- [4] M.X. Goemans and D.J. Bertsimas, Survivable networks, linear programming relaxations and the parsimonious property. *Math. Program.* **60** (1993) 145–166.
- [5] Y.J. Chu and T.H. Liu, On the shortest arborescence of a directed graph. *Scientia Sinica* **14** (1965).
- [6] J. Könemann, G. Konjevod, O. Parekh and A. Sinha, Improved approximations for tour and tree covers. *Algorithmica* **38** (2003) 441–449.
- [7] D.B. Shmoys and D.P. Williamson, Analyzing the help-karp tsp bound: a monotonicity property with application. *Inf. Process. Lett.* **35** (1990) 281–285.
- [8] V.V. Vazirani and S. Rajagopalan, On the bidirected cut relaxation for metric bidirected steiner tree problem, in *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms* (1999) 742–751.
- [9] L.A. Wolsey, Heuristic analysis, linear programming and branch-and-bound. *Math. Program. Stud.* **13** (1980) 121–134.