As appearing in *Information Processing Letters*, 18 October 1993.

APPROXIMATING THE TREE AND TOUR COVERS OF A GRAPH

Esther M. Arkin[†]
Department of Applied Mathematics and Statistics
SUNY Stony Brook
Stony Brook, NY 11794-3600
estie@ams.sunysb.edu

Magnús M. Halldórsson
School of Information Science
Japan Advanced Institute of Science and Technology – Hokuriku
Tatsunokuchi, Ishikawa 923-12, Japan
magnus@jaist-east.ac.jp

Refael Hassin
Department of Statistics and Operations Research
School of Mathematical Sciences
Tel-Aviv University
Tel-Aviv 69978, Israel
hassin@math.tau.ac.il

April 11, 1994

Abstract

The *tree* and *tour cover* problems on an edge-weighted graph are to compute a minimum weight tree and closed walk, respectively, whose vertices form a vertex cover. Both problems are NP-hard. In this note we give strongly polynomial time, constant factor approximation algorithms for both problems. An interesting feature of our algorithms is how they combine approximations of other problems, namely the weighted vertex cover, traveling salesman, and Steiner tree problems.

[†]Partially supported by NSF Grants CCR-9204585 and ECSE-8857642.

1 Introduction

Our starting point is the weighted vertex cover problem: Given a graph $G = (V, E)$ with weights on the vertices find a minimum weight set of vertices that "cover" all edges, i.e., a set of vertices $V' \subseteq V$ such that for each edge $\{u, v\} \in E$ at least one of u and v belongs to V'. This problem is NP-hard [GJ], however it can be approximated by a cover whose weight is at most double the optimal weight [Ho, BE] (see [Mo] for an updated survey of bounded approximations for the problem).

In this note we are concerned with a class of problems that generalizes the vertex cover problem. A subgraph $G' = (V', E')$ $V' \subset V$, $E' \subset E$ is said to cover $G = (V, E)$ if V' is a vertex cover of the edges of the graph G. Given a connected graph $G = (V, E)$ with weights on the edges, we wish to find a minimum weight subgraph from a given class that covers G.

Specifically, we describe approximation algorithms with bounded error ratios for two cases. In the *tree cover problem* the subgraph is required to be connected, and thus, in the case of nonnegative weights it will be a tree. In the *tour cover problem* the subgraph is required to be a closed walk. Note that in our problems the weights are associated with edges, not vertices. Also, we do not assume that the edge weights satisfy the triangle inequality.

Both problems are clearly NP-hard. The minimum weight tree cover is hard even when all edge weights are equal, since it is asking for a minimum connected vertex cover ([GJ], page 190). A simple reduction from the *traveling salesman problem*, TSP, shows the hardness of the tour cover problem. (For each vertex in an instance of the TSP, add an edge whose one endpoint is that vertex and the other endpoint is a vertex of degree 1.) On the other hand, as mentioned above, the vertex cover can be approximated by a cover whose weight is at most double the optimum weight, and the TSP can be approximated by a tour whose length is at most 3/2 times the optimum length, if the triangle inequality is assumed [Ch]. (Since we are looking for a closed walk, which may repeat vertices, not assuming the triangle inequality is not a problem.) Thus we are interested in finding approximation algorithms to the tree and tour cover problems.

Another motivation of our study is the fact that the minimum weight tour cover can also be viewed as a watchman route problem, in which the watchman must see every edge by visiting at least one of its endpoints. The geometric version of the watchman route problem is: Find a shortest path (or cycle) for a watchman in an art gallery, that allows the watchman to "see" all of the gallery. The resulting problem is called the *watchman route problem*, see for example [CN]. To date no approximation methods are known for the watchman route problem, not even ones that obtain a performance ratio of $O(\log n)$ for an n-sided polygonal environment.

The tour cover problem is a special case of the *traveling purchaser problem* (TPP) [Ong]. In this problem we are given in addition to the weighted graph G also a set I of items and a cost matrix describing the cost of purchasing item $i \in I$ at each vertex $v \in V$. (We can assume that each item is available at all vertices by placing a very large cost for the item at vertices in which it is not available.) The problem is to compute a tour such that the total costs of travel and purchasing (each item at the least expensive vertex on the tour) is minimized. To obtain the tour cover problem as a special case, define for each edge $\{j, k\}$ an item that can be purchased only at vertices j and k for a unit cost, while its cost elsewhere is very large. The TPP has both the TSP and the set cover as immediate special cases. The TSP is obtained by defining a unique item for each vertex, available only at that vertex (or equivalently, available at other vertices at a high cost). In the set cover problem subsets $S_1, ..., S_n \subset I$ with weights $w_1, ..., w_n$ are given. It is obtained as a special case of TPP by defining the purchase costs at $i \in V$ to be zero for S_i and infinity for $I \setminus S_i$, and letting the length of edge $\{i, j\}$ be $\frac{w_i + w_j}{2}$. We also note that the TPP generalizes the prize collecting TSP studied by [BGSW]. It follows from [BGLR] that unless $P = NP$ no bounded approximation for set cover, and hence for TPP exists. Furthermore, an approximation of factor $c \log n$ (for any constant $c < 1/8$) can not be achieved unless NP is contained in $DTIME(n^{\log \log n})$.

Both vertex cover and TSP are known to have a lower bound of a fixed constant performance ratio achievable in polynomial time (assuming $P \neq NP$) [ALMSS]. The reduction above shows the bound applies to the tour cover problem. The following (similar) reduction from vertex cover shows the bound for the tree cover problem: Add a new vertex v adjacent to all other vertices, and let all edge weights be equal. A solution to the tree cover problem, which is a connected vertex cover on the new graph, minus the new vertex v, corresponds to a solution to the vertex cover of the original graph.

A related class of problems deals with the location of path-shaped or a tree-shaped facilities on a graph

[AN, HSL, KLM, KLWF, Mi, R]. There, a subgraph with a given property is looked for such that the maximum distance between any vertex of the graph and its closest vertex of the subgraph is minimized. The "cover version" of the problem, where a maximum allowed distance (a radius) is given, and the size of the subgraph is to be minimized, is especially close to our problem. There, it is desired to cover vertices while in our problem we cover edges. Using our terminology, the above radius problem could be named "the dominating subgraph problem" (though some of the authors of the abovementioned references use the term "covering subgraph").

We note that the tour and tree problems refer to the same setting but with a different measure of distance. In the tree version repeating a segment for the second time is free of charge. One could picture an application of the tree measure in a situation where a tunnel must be dug or a mine field must be crossed. The objective is then to minimize the total length to be dug or to be freed of mines.

The remainder of this paper is organized as follows: In Section 2 (resp. 3) we give our approximation to the tree (resp. tour) cover problem, and we conclude with open problems and discussion.

2 Approximating the tree cover

2.1 Weighted Tree Cover

We begin with a formal definition of the problem:

Problem 2.1 (Tree Cover Problem) *Given a graph $G = (V, E)$ and weight on the edges $w_e \geq 0$ for all $e \in E$, find a minimum weight tree, $T_1 \subseteq E$, whose vertices cover all the edges E.*

Before describing our approximation algorithm, we show that (not surprisingly) a simple modification of Prim or Kruskal's minimum spanning tree algorithms does not yield a constant factor approximation. The modification is a change to the stopping criteria of the algorithms: Continue adding edges to the tree cover as long as there is an edge of the original graph which is not covered and the resulting graph is connected. Consider the following graph $G = (V, E)$, where $V = \{a, b, c_1, c_2, \ldots, c_m\}$, and the edges are (a, c_i) of weight 1, (b, c_i) of weight 2, for all $1 \leq i \leq m$, and edge (a, b) of weight 3. Clearly the optimal tree cover is the single edge (a, b) of weight 3, but both the modified Prim and Kruskal algorithms will return the tree cover containing all edges (a, c_i) whose total weight is m.

Our algorithm has two main steps: Find a set of edges whose vertices form a vertex cover, and connect these edges into a tree. We use the following contraction operation before searching for a set of connecting edges.

Definition 2.2 *The* contraction *of a graph G along a set of edges Y produces a graph $G' = (V', E')$ and a set of vertices $S \subseteq V'$ defined as follows: For each edge in Y, merge its two endpoints into a single vertex, whose adjacency list is the union of the two adjacency lists. If parallel edges occur, retain the edge with smaller weight. The resulting graph is G'. The set S consists of the vertices formed by edge contraction (i.e., the nodes in $V' \setminus V$).*

We now state our algorithm, which is really a scheme of algorithms parametrized by k and the algorithms used to approximate the weighted vertex cover and Steiner tree in steps 3 and 4:

1. Let Q_1 be the best solution on at most k vertices, if one exists, and E otherwise.
2. Define a weight for each vertex i, to be the minimum weight of any edge touching it, $\hat{w}_i = min\{w_e \mid e = (i, j) \in E\}$.
3. Find an approximate weighted vertex cover VC on G with vertex weights $\hat{w}_i$. Denote by EC the set of edges that yield the weights $\hat{w}_i$, for nodes $i \in VC$. In other words, EC is the set of edges obtained by every vertex in VC "selecting" a shortest edge incident to it.
4. Contract G along edges EC to obtain G' and S. Find an approximation to the Steiner tree in G' with terminals S, and map this solution back to G.

5. Let Q_2 be the union of the Steiner tree approximation and the edges EC obtained from the vertex cover approximation. Output the smaller of the two solutions Q_1 and Q_2. Let APX_1 denote the length of the approximate solution obtained.

First observe that the problem can be solved efficiently if the number of vertices in a solution is small.

Lemma 2.3 *The problem of either finding a tree cover with at most k vertices of minimum weight or determining that no such solution exists, can be solved in time $O(|E| + k^{2k+3})$.*

Proof: The vertices of a tree cover form a vertex cover. It follows that a solution with at most k vertices must contain every vertex whose degree is greater than k. We therefore add all such vertices to our prospective solution, and remove the vertices, incident edges, and possible isolated vertices from the graph. The remaining graph can contain at most $k(k-1)$ vertices if there is to be a cover with only k vertices. It then suffices to check all subsets of size at most k. ∎

Denote by OPT_{WVC} (APX_{WVC}) the weight of the optimal (approximate) weighted vertex cover of the input graph G. Let T_1 be an optimal tree cover, OPT_1 its weight, and $t_1 > 0$ the number of vertices in T_1. We have:

Lemma 2.4 $\mathrm{OPT}_{WVC} \leq (1 + \frac{1}{t_1 - 1})\mathrm{OPT}_1$.

Proof: Consider the optimal tree T_1. Let the root r of the tree be a vertex of minimum weight in the tree $\hat{w}_r = \min\{\hat{w}_i \mid i \in T_1\}$. Direct all edges in the tree to go towards the root. Define $\hat{w}'_i$ to be the weight of the edge in the tree leaving vertex i, and $\hat{w}'_r = 0$. (Note that by our choice of edge direction, this definition is unambiguous.) Clearly, $\mathrm{OPT}_1 = \sum_{i \in T_1} \hat{w}'_i$, and for every vertex in the tree other than the root, r, $\hat{w}'_i \geq \hat{w}_i$. Also, since T_1 is a tree cover, its vertices are a vertex cover of G. We use the notation $\hat{w}(T_1) = \sum_{i \in T_1} \hat{w}_i$, etc. Thus:

$$\mathrm{OPT}_1 \;=\; \hat{w}'(T_1) \;=\; \hat{w}(T_1) - \hat{w}_r \;\geq\; \hat{w}(T_1)(1 - 1/t_1) \;\geq\; \mathrm{OPT}_{WVC}(1 - 1/t_1).$$

∎

The bound given by Lemma 2.4 can be attained, as can be seen from a simple example. Consider a graph whose vertices are four points on a line with coordinates $0, 1, 2, 3$, and the edges are $(0,1)$, $(1,2)$ and $(2,3)$ all of length 1. The optimal tree cover T_1 contains the middle edge $(1,2)$ and $\mathrm{OPT}_1 = 1$, $t_1 = 2$, while $\mathrm{OPT}_{WVC} = 2$.

Let $\mathrm{OPT}_{St}(G', S)$ ($\mathrm{APX}_{St}(G', S)$) denote the weight of an optimal (approximate) Steiner tree of the terminals S in the contracted graph G'. Let r_{St} (r_{WVC}) be the performance ratio obtained by the algorithm used to approximate the Steiner tree (weighted vertex cover).

Theorem 2.5 *The tree cover found by the algorithm above has weight at most $r_{St} + r_{WVC}(1 + \frac{1}{k})$ times the optimal solution.*

Proof: Our solution is the union of two edge sets, thus

$$\begin{aligned} \mathrm{APX}_1 &= w(EC) + \mathrm{APX}_{St}(G', S) \\ &\leq \mathrm{APX}_{WVC} + \mathrm{APX}_{St}(G', S) \\ &\leq r_{WVC} \cdot \mathrm{OPT}_{WVC} + r_{St} \cdot \mathrm{OPT}_{St}(G', S). \end{aligned}$$

The first inequality holds since the weight of every edge in EC is counted by at least one of its endpoints in VC. Since each terminal of the network (G', S) is a contraction of one or more edges in which at least one endpoint is contained in and connected by the optimal tree cover, $\mathrm{OPT}_{St}(G', S) \leq \mathrm{OPT}_1$. It now follows from lemma 2.4 that:

$$\mathrm{APX}_1 \;\leq\; (r_{WVC}(1 + \tfrac{1}{t_1 - 1}) + r_{St})\ \mathrm{OPT}_1.$$

If the optimal tree cover contains k or fewer vertices, our heuristic solution will be optimal. Thus we may assume that $t_1 \geq k + 1$. ∎

The complexity of our approach depends on the algorithms used for the weighted vertex cover and Steiner tree approximations, and to a lesser extent the value of k. In fact, we have a range of choices that provide a performance/complexity tradeoff.

For the weighted vertex cover problem, a 2-approximation can be found in linear time, while a $2 - \log\log|V|/2\log|V|$ approximation can be obtained in time $O(|V||E|)$ [BE]. For a Steiner tree we can obtain:

- 2-approximation in time $O(|E| + |V|\log|V|)$ [Me],
- 11/6-approximation in time $O(|S|(|E| + |V||S| + |V|\log|V|))$ [Ze], and
- 16/9-approximation in time $O(|V|^2|S|^3 + |V|^3)$ [BR].

In particular, our approach can obtain a $34/9 < 3.78$ performance ratio in time $O(|V|^2|S|^3 + |V|^4)$, by using the Steiner tree algorithm of [BR] and the more powerful vertex cover algorithm of [BE], and by applying lemma 2.3 with $k = 2\log n/\log\log n$. Alternatively, we can find a tree cover of weight at most $4 + O(\log\log n/\log n)$ times the optimum in time $O(|E| + |V|\log|V|)$.

2.2 Unweighted Tree Cover

If the underlying graph is unweighted we can obtain better bounds on the approximation ratios. The algorithm largely stays the same, with weights being unit weights, step 1 checking only for a trivial single-node solution, and step 3 changed to:

3. Find an approximate vertex cover VC on G. Let EC be the edges of a spanning forest of VC.

Theorem 2.6 *Unweighted tree cover can be approximated within a factor of 3 in linear time.*

Proof: We use a maximal matching M as our vertex cover approximation. Let $t = |M|$. If EC is connected, we output $Q_2 = EC$. Otherwise, (EC is not connected) we let Q be any minimal connected subgraph spanning S, and output $Q_2 = Q \cup M$.

The optimal cover must contain at least one vertex of each edge of M and must connect them together. If it doesn't use any additional vertices, neither will our solution, in which case we use $2t - 1$ edges versus optimal solution of $t - 1$. Since we may assume $t \geq 2$, the ratio is at most 3.

On the other hand, we never use more than $3t - 2$ edges since at most one vertex can link two terminals in the MST solution. Thus, if the optimal solution contains an additional vertex, it has at least t edges. Combined, we obtain a ratio of $(3t - 2)/t \leq 3$. ∎

Better yet, we can get within a factor of 2, which is best possible in the sense that a better constant would directly lead to an improved performance ratio for the thoroughly studied vertex cover problem.

Theorem 2.7 *Unweighted tree cover can be approximated within a factor of 2 in polynomial time.*

Proof: For a vertex cover approximation we use the set $C \cup W$, where C is a maximal collection of disjoint odd cycles in G and W is an optimal vertex cover of the remaining bipartite graph $G \setminus C$. This method is closely related to the vertex cover algorithms of Bar-Yehuda and Even [BE] and Monien and Speckenmeyer [MS].

Let t be the number of odd cycles found, $|C_i|$ be the number of vertices in the i-th cycle, and $|C|$ the total number of vertices in the cycle collection.

Since $C \cup W$ forms a vertex cover, at least one endpoint of every edge is in $C \cup W$. Thus, to connect the cycles of C and the vertices of W, at most $t + |W| - 1$ additional vertices must suffice. Hence, the number of vertices in the solution is at most $2|W| + |C| + t - 1$, and the number of edges precisely one less or

$$\mathrm{APX}_1 \;\leq\; 2|W| + |C| + t - 2.$$

On the other hand, any solution must contain at least $p+1$ vertices from any odd cycle of length $2p+1$, as well as $\mathrm{OPT}_{VC}(G \setminus C) = |W|$ vertices from the remaining part of the graph. Hence, the optimal number of vertices in a tree cover is at least $|W| + \sum_i (|C_i| + 1)/2$, and the number of edges

$$\mathrm{OPT}_1 \;\geq\; |W| + \sum_i (|C_i| + 1)/2 - 1 \;=\; |W| + (|C| + t)/2 \;\geq\; \frac{\mathrm{APX}_1}{2}.$$

■

A maximal collection of disjoint odd cycles can be found in $O(|V||E|)$ time by running breadth-first search starting at each vertex. An optimal vertex cover of a bipartite graph can be found using matching techniques (see [La, p.190] for a theorem of König-Egervary) in time $O(|E|\sqrt{|V|})$ [HK]. The connecting vertices can easily be found in linear time. Hence, the combined time complexity is $O(|V||E|)$.

We can also approximate the following interesting generalization of the problem:

Problem 2.8 (Generalized Unweighted Tree Cover Problem) *Given a graph $G = (V, E)$ and a set of edges E' where $E' \subseteq E$, find a tree, $T_1 \subseteq E$, with minimum number of edges whose vertices cover all the edges E'.*

Theorem 2.9 *Using the vertices of a maximal matching of E' as a vertex cover approximation in step 3, the performance ratio of our algorithmic scheme for the generalized unweighted problem is at most $1 + r_{St}$.*

Proof: Let t be the number of edges in the maximal matching of E'. If we assume that only the edges of the matching are contracted, then t equals $|S|$, the number of terminals in the contracted network. If further edges are contracted, it can only help in reducing the costs of connecting the edges of the matching into a (Steiner) tree. Also, since an optimal tree cover must contain a point in each edge of the matching and must connect them together, $\mathrm{OPT}_1 \geq \mathrm{OPT}_{St}(G', S)$. Hence,

$$\frac{\mathrm{APX}_1}{\mathrm{OPT}_1} \leq \frac{|S| + \mathrm{APX}_{St}(G', S)}{\mathrm{OPT}_{St}(G', S)} \equiv r_1. \tag{1}$$

The number of edges of an optimal Steiner tree is always at least one fewer than the number of terminals $|S|$, but when equality holds the Steiner problem reduces to finding a spanning tree of the subgraph induced by $|S|$. Thus, we may assume that $|S| \leq \mathrm{OPT}_1$, and hence equation (1) yields $r_1 \leq 1 + r_{St}$. ■

In particular, a ratio of $r_1 \leq 25/9 < 2.78$ is possible using the Steiner tree algorithm of [BR]. Observe that r_1 to be minimized in equation (1) is of a special nature, and in fact a bound $r_1 \leq 2.46$ has been shown [Ha].

3 Approximating the tour cover

The problem we discuss in this section is a modification of the previous one, asking for a tour instead of tree cover. We use the word "tour", although we allow repeated vertices and edges.

Problem 3.1 (Tour Cover Problem) *Given a graph $G = (V, E)$ and weight on the edges $w_e \geq 0$ for all $e \in E$, find a minimum weight closed walk, $T_2 \subseteq E$, whose vertices cover all the edges E.*

One possible approach for the design of an approximation algorithm is to use an approximated tree cover, and then add to it either a matching or a duplicate of the tree cover, to obtain a tour. Our method is somewhat different and yields better approximation bounds.

Our algorithm has five steps where the first three are identical to the weighted tree cover algorithm, except the first step need only check if a trivial single-node solution exists. The last two steps are replaced by:

4. Find a TSP approximation of the contracted graph, with distances modified (if necessary) to the shortest paths distances. Map this solution back to a partial tour Q of G.

5. For each component formed by the edges in EC, form an Eulerian walk from the entry point to the exit point of Q in the component. Output the tour formed by Q and the Eulerian walks, and let APX_2 denote the length of the approximate solution obtained.

Let T_2 be an optimal tour cover, OPT_2 its weight, and t_2 the number of vertices in the optimal tour cover. We have:

Lemma 3.2 *If $t_2 > 1$ then* $\mathrm{OPT}_{WVC} \leq \mathrm{OPT}_2$.

Proof: The proof is very similar to that of Lemma 2.4. (In fact it is easier, since we do not have to treat the root separately.) Consider the optimal tour T_2. Arbitrarily direct all the edges of the tour to go around the tour in one of the two possible directions. Define $\hat{w}'_i$ to be the weight of the edge in the tour leaving vertex i. (Note that by our choice of edge direction, this definition is unambiguous.) Clearly, $\mathrm{OPT}_2 = \sum_{i \in T_2} \hat{w}'_i$, and for every vertex in the tour, $\hat{w}'_i \geq \hat{w}_i$. Also, since T_2 is a tour cover, it contains a vertex cover of all edges, which we denote by Q. Hence,

$$\mathrm{OPT}_{WVC} \;\leq\; \hat{w}(Q) \;\leq\; \hat{w}'(Q) \;\leq\; \hat{w}'(T_2) \;\leq\; \mathrm{OPT}_2.$$

■

The bound stated by Lemma 3.2 can be attained. This can be demonstrated by the same four points example as in Lemma 2.4, where $\mathrm{OPT}_2 = \mathrm{OPT}_{WVC} = 2$.

Let $\mathrm{OPT}_{TSP}(G', S)$ ($\mathrm{APX}_{TSP}(G', S)$) denote the weight of an optimal (an approximate) traveling salesman tour on the vertices S inside graph G'. Since edge weights in G' are the lengths of the shortest paths, the weights satisfy the triangle inequality. Let r_{TSP} be the constant factor by which the TSP algorithm given approximates the length of a tour, so $\mathrm{APX}_{TSP}(G', S) \leq r_{TSP}\mathrm{OPT}_{TSP}(G', S)$. We know that assuming symmetric distances, and using the fact that the triangle inequality holds, it is possible to get $r_{TSP} = \frac{3}{2}$ [Ch].

Theorem 3.3 *The tour cover found by the algorithm above has weight at most $2r_{WVC} + r_{TSP}$ times the optimal solution.*

Proof: There is an Eulerian walk between any two vertices in a connected graph that uses each edge at most twice. To see that, form a multigraph where every edge is repeated, and subtract from it one copy of some path between the two vertices. Each vertex has an even degree except the starting and the ending vertex, hence by the oldest theorem in graph theory, there is a walk that traverses every edge of the multigraph. It follows from this that

$$\mathrm{APX}_2 \;\leq\; 2 \cdot \mathrm{APX}_{WVC} + \mathrm{APX}_{TSP}(G', S).$$

By Lemma 3.2, $\mathrm{OPT}_{WVC} \leq \mathrm{OPT}_2$, and since every component must be connected by the optimal tour, $\mathrm{OPT}_{TSP}(G', S) \leq \mathrm{OPT}_2$. The theorem now follows. ■

Finally, we analyze the running time of the algorithm: The first three steps of the algorithm are as for the tree cover problem. The fourth step in which we use an approximation of the TSP is the most time consuming part. To achieve the $r_{TSP} = 1.5$ bound for the case of symmetric distances (and hence 5.5 overall) we need $O(|V|^3)$ time.

An Eulerian walk of a minimum spanning tree of the reduced graph is a walk of all the vertices of cost at most twice the optimal TSP solution, for a overall ratio of 6. This can be obtained in time $O(|E| + |V| \log |V|)$ since MST computation does not require the triangle inequality, saving the expensive distance graph computation.

For the unweighted problem, we can argue as in the previous section that a maximal matching as a vertex cover approximation costs at most the length of a tour cover. This saves a factor of 2, for a ratio of $2 + r_{St}$ in time proportional to the complexity of the TSP heuristic. In particular, this implies a ratio of 3.5 in time $O(|V|^3)$ and a ratio of 4 in linear time. The same bounds hold for the the generalized unweighted problem, where we are given a subset of the edges to be covered.

4 Open Problems and Discussion

A question arises from the proof of Theorems 2.5 and 3.3, in which we use the fact that an optimal tree or tour cover must connect all connected components of EC. This is due to the fact that we require our cover to cover all edges of the graph. If the problem asks for a cover that covers edges $E' \subset E$ our proof is not valid, and in fact we do not know of a combinatorial approximation algorithm for the weighted version with a guaranteed bound.

An alternative approach approximating the tour cover was suggested by Bienstock and Simchi-Levi (private communications) similar to the one used in [BGSW] to approximate a prize collecting traveling salesman tour. The approach uses an LP relaxation of the problem, and careful rounding of the solution. This method is not as fast as the one we propose, but may yield lower error bounds.

Acknowledgement

We wish to thank David Shmoys and Jaikumar Radhakrishnan for helpful discussions.

References

[AN] Y. P. Aneja and K.P.K. Nair, "Location of a tree shaped facility in a network", *INFOR* **30**, 319–324, 1992.

[ALMSS] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy, "Proof verification and hardness of approximation problems", *Proc. 33rd IEEE Symposium on Foundations of Computer Science*, Pittsburgh, PA, 14–23, 1992.

[BE] R. Bar-Yehuda and S. Even, "A local-ratio theorem for approximating the weighted vertex cover problem", *Annals of Disc. Math.*, 25:27–44, 1985.

[BGLR] M. Bellare, S. Goldwasser, C. Lund and A. Russel, "Efficient probabilistically checkable proofs: Applications to approximations", *Proc. 25th ACM Symposium on the Theory of Computing*, 294–304, 1993.

[BR] P. Berman and V. Ramaiyer, "Improved approximations for the Steiner tree problem", *Proc. Third Annual ACM-SIAM Symposium on Discrete Algorithms*, Orlando, FL, January, 325–334, 1992.

[BGSW] D. Bienstock, M.X. Goemans, D. Simchi-Levi, and D.P. Williamson, "A note on the prize collecting traveling salesman problem", to appear *Math. Prog.* 1993.

[CN] W.P. Chin and S. Ntafos, "Optimum watchman routes", *Inform. Process. Lett.* **28**, 39–44, 1988.

[Ch] N. Christofides, "Worst-case analysis of a new heuristic for the traveling salesman problem", Technical Report, GSIA, Carnegie-Mellon University, 1976.

[GJ] M. R. Garey, and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, 1978.

[HSL] S.L. Hakimi, E.F. Schmeichel and M. Labbé, "On locating path- or tree-shaped facilities on networks ", manuscript, 1990.

[Ha] M. M. Halldórsson, unpublished, 1993.

[Ho] D.S. Hochbaum, "Approximation algorithms for the set covering and vertex cover problems", *SIAM J. Computing*, **11**, 555–556, 1982.

[HK] J.E. Hopcroft, and R.E. Karp, "An $n^{5/2}$ algorithm for maximal matchings in bipartite graphs", *SIAM J. Computing*, **4**, 225–231, 1973.

[KLM] R.K. Kincaid, T.J. Lowe and T.L. Morin, "The location of central structures in trees", *Comput. Opns. Res.* **15**, 103–113, 1988.

[KLWF] T.U. Kim, T.J. Lowe, J.E. Ward and R.L. Francis, "A minimum length covering subgraph of a network", *Annals of Operations Research* **18**, 245–260, 1989.

[La] E. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, 1976.

[Me] K. Mehlhorn, "A faster approximation algorithm for the Steiner problem in graphs", *Inform. Process. Lett.*, **27**, 125–128, 1988.

[Mi] E. Minieka, "The optimal location of a path or tree in a tree network", *Networks* **15**, 1985.

[MS] B. Monien and E. Speckenmeyer, "Ramsey numbers and an approximation algorithm for the vertex cover problem", *Acta Inf.*, 22:115–123, 1985.

[Mo] R. Motwani, "Lecture notes on approximation algorithms", Volume I, Stanford University, 1992.

[Ong] H.L. Ong, "Approximate algorithms for the traveling purchaser problem", *Operations Research Letters* **1**, 201–205, 1982.

[R] M. B. Richey, "Optimal location of a path or tree on a network with cycles", *Networks* **20**, 391–407, 1990.

[Ze] A. Z. Zelikovsky. "A faster approximation algorithm for the Steiner tree problem in graphs", *Inform. Process. Lett.*, **46**, 79–83, 1993.