# The Multi-Integer Set Cover and the Facility Terminal Cover Problem

**Dorit S. Hochbaum**
*Department of Industrial Engineering and Operations Research and Walter A. Haas School of Business, University of California, Berkeley, California 94720*

**Asaf Levin**
*Department of Statistics, The Hebrew University, Jerusalem 91905, Israel*

**The *facility terminal cover* problem is a generalization of the vertex cover problem. The problem is to "cover" the edges of an undirected graph $G = (V, E)$ where each edge $e$ is associated with a non-negative demand $d_e$. An edge $e = [u, v]$ is covered if at least one of its endpoint vertices is allocated capacity of at least $d_e$. Each vertex $v$ is associated with a non-negative weight $w_v$. The goal is to allocate capacity $c_v \geq 0$ to each vertex $v$ so that all edges are covered and the total allocation cost, $\sum_{v \in V} w_v c_v$, is minimized. A recent paper by Xu et al. [Networks 50 (2007), 118-126], studied this problem, and presented a $2e$- approximation algorithm for this problem for $e$ the base of the natural logarithm. We generalize here the facility terminal cover problem to the *multi-integer set cover*, and relate that problem to the set cover problem, which it generalizes, and the multi-cover problem. We present a $\Delta$-approximation algorithm for the multi-integer set cover problem, for $\Delta$ the maximum *coverage*. This demonstrates that even though the multi-integer set cover problem generalizes the set cover problem, the same approximation ratio holds. In the special case of the facility terminal cover problem this yields a 2-approximation algorithm, and with run time dominated by the sorting of the edge demands. This approximation algorithm improves considerably on the result of Xu et al.**
 **NETWORKS, Vol. 53(1), 63–66 2009**

**Keywords:** algorithms; approximation algorithms; vertex cover; set cover; network design

## 1. INTRODUCTION

The *facility terminal cover* problem (FTC) is defined as follows. Given an undirected graph $G = (V, E)$ where each vertex $v$ is associated with a non-negative weight $w_v$, and each edge $e$ is associated with a non-negative demand $d_e$, the goal is to allocate capacity $c_v \geq 0$ to each vertex $v \in V$ so that for every edge $e = [u, v]$ either $c_u \geq d_e$ or $c_v \geq d_e$, and so that $\sum_{v \in V} w_v c_v$ is minimum. We note that it is possible to allocate no capacity to a vertex $v$ by setting $c_v = 0$. The problem FTC was introduced by Xu et al. [11], and it was motivated by several problems in networks and operations research.

For an algorithm $\mathbf{A}$, denote the objective value of a solution it delivers on an input $I$ by $\mathbf{A}(I)$. An optimal solution is denoted by OPT, and the optimal objective value is denoted by OPT as well. The (absolute) approximation ratio of $\mathbf{A}$ is defined as the infimum $\rho$ such that for any input $I$, $\mathbf{A}(I) \leq \rho \cdot \mathrm{OPT}(I)$. We restrict ourselves to algorithms that run in polynomial time.

The *vertex cover* problem is a special case of FTC where the demand of each edge is one, i.e., for all $e \in E$ $d_e = 1$. For the vertex cover problem any optimal solution consists of capacity allocation to the vertices that is either zero or one. For the FTC problem the capacity can assume the value zero or any value in the set $\{d_e | e \in E\}$.

Previous results for the weighted vertex cover problem include the first 2-approximation algorithms by [6] that was shown to run in the time of solving a minimum cut on a bipartite graph in [7]. Various algorithms with the same approximation ratio, or slightly better for restricted types of graphs, were shown, for example, in [1,7,9]. In [7] it was conjectured that unless P = NP it is impossible to improve the approximation ratio to $2-\varepsilon$ for any $\varepsilon > 0$ for the vertex cover problem on general graphs. To date NP-hardness results confirmed that unless P = NP it is impossible to approximate the vertex cover problem within $10\sqrt{5} - 21 \simeq 1.3606$ [3], and under the stronger assumption of the unique games conjecture it is impossible to approximate the vertex cover problem within an approximation ratio of $2 - \varepsilon$ for all $\varepsilon > 0$ [10]. A survey of approximation algorithms for the vertex cover problem and the set cover problem is presented in [8].

Received August 2007; accepted January 2008
*Correspondence to*: A. Levin; e-mail: levinas@mscc.huji.ac.il
Contract grant sponsor: NSF; Contract grant numbers: DMI-0620677
DOI 10.1002/net.20265
Published online 13 August 2008 in Wiley InterScience (www.interscience.wiley.com).

Another related problem is *vertex cover with hard capacities*, which is the variant of vertex cover where each selected vertex can be used to cover up to a prespecified number of edges incident to this vertex (its capacity). Chuzhoy and Naor [2] proved that the weighted problem is at least as hard (to approximate) as the set cover problem, and for the unweighted version they presented a 3-approximation algorithm. This 3-approximation algorithm was improved by Gandhi et al. [4] to a 2-approximation algorithm.

Xu et al. [11] presented a $2e$-approximation algorithm for FTC where $e$ is the base of the natural logarithm. Their algorithm is based on randomized geometric grouping of the edges into sets of roughly equal demand edges, and then applying a vertex cover 2-approximation algorithm for each resulting edge class. They also wrote that "It also seems difficult to apply those LP-based methods to solve the FTC problem, as the natural formulation of the FTC problem is a quadratic programming."

Our contributions here are in several respects: We present a 2-approximation algorithm for FTC. We present a formulation of FTC as a *linear* mixed integer programming problem. A linear programming relaxation of this formulation is used to obtain the 2-approximation algorithm. We devise an alternative $O(m \log n)$ time 2-approximation algorithm for FTC based on the primal-dual scheme. We then define a generalization of the set cover problem, the FTC problem and the multi-cover problem, which we call the *multi-integer set cover problem* or MISC. We provide a mixed integer programming formulation of MISC and generate from the relaxation, or from a primal-dual approach, an efficient approximation algorithm for the problem that gives the same approximation bound as for the easier set cover problem.

The *multi-integer set cover* problem (MISC) is a generalization of FTC for hypergraphs and a generalization of the set cover problem. In a graph each edge is a set of 2 endpoint vertices, whereas in a hypergraph a "hyperedge" is an arbitrary sized set of "vertices". A formal definition of MISC, in which sets are to be covered and which is "dual" to another possible formalization, is as follows: Given a collection of sets (the hyperedges) $S_1, S_2, \ldots, S_m$ over a ground set $V$, where each element $v \in V$ is associated with a non-negative weight $w_v$, and each set $S_i$ is associated with a non-negative demand $d_i$. The goal is to allocate capacities to the elements, $c_v \geq 0$ is the capacity of $v$, such that for each set $S_i$ there is an element $u \in S_i$ with $c_u \geq d_i$ and so that $\sum_{v \in V} w_v c_v$ is minimized. Clearly, MISC generalizes FTC where each set $S_i$ consists of the two end-vertices of the edge $e_i$ in the graph $G$.

The following notation is used throughout: Vectors are denoted in boldface, so $\mathbf{x}$ is a vector and $x_i$ is the value of the ith entry in the vector. $\Delta$ denotes the *maximum coverage*, $\Delta = \max_{i=1,2,\ldots,m} |S_i|$. For FTC $\Delta = 2$. $A^{(1)}$ denotes the total size of the hyperedges in the input, $A^{(1)} = \sum_{i=1}^{m} |S_i|$. Obviously $A^{(1)} \leq \Delta m$. We let the size of $V$ be $n = |V|$.

In Section 2 we present a $\Delta$-approximation algorithm for MISC based on solving a linear program. In Section 3 we present a $\Delta$-approximation algorithm for MISC using the primal-dual technique with an improved time complexity of $O(m \log n + A^{(1)})$. Our approximation algorithm gives a 2-approximation algorithm for FTC thus improving the result of [11].

## 2. FORMULATION AND A LINEAR PROGRAMMING APPROXIMATION ALGORITHM

Let $x_v$ be a decision variable indicating the capacity assigned to $v \in V$, and $z_{iv}$ be a binary variable for each pair $(i, v)$, $i \in \{1, \ldots, m\}$ and $v \in S_i$, indicating whether element $v \in S_i$ is designated to cover $i$ and thus assume a capacity level at least $d_i$:

$$
\begin{array}{lllll}
 & \min & \sum_{v \in V} w_v x_v & & \\
 & \text{subject to} & \sum_{v \in S_i} z_{iv} \geq 1 & i = 1, \ldots m & \\
\text{(MISC)} & & x_v \geq d_i z_{iv} & i = 1, \ldots m & v \in V \\
 & & 0 \leq z_{iv} & \text{integer for } 1 = 1, \ldots, & \\
 & & & m \text{ and } v \in V. &
\end{array}
$$

In this formulation it is not necessary to restrict the value of the variables $z_{iv}$ to be $\leq 1$. It is easy to see that for any feasible MISC problem there is always an optimal solution where the variables $z_{iv}$ are binary.

A fractional feasible solution to (MISC) is obtained by solving the linear programming (LP) relaxation of the problem:

$$
\begin{array}{llll}
 & \min & \sum_{v \in V} w_v x_v & \\
\text{(LP-MISC)} & \text{subject to} & \sum_{v \in S_i} x_v \geq d_i & i = 1, \ldots m \\
 & & 0 \leq x_v & \text{for all } v \in V.
\end{array}
$$

Obviously an optimal solution of value OPT to MISC is a feasible solution to (LP-MISC) and with the same objective value. To see that LP-MISC is indeed a relaxation of MISC notice that for an optimal solution $\bar{\mathbf{x}}$ to LP-MISC, setting $z_{iv} = \bar{x}_v / d_i$ is a feasible fractional solution to MISC.

We conclude that the objective value of an optimal solution $\bar{\mathbf{x}}$ to LP-MISC is at most OPT. We now argue that the solution $\Delta\bar{\mathbf{x}} = (\Delta\bar{x}_v)_{v \in V}$ resulting from multiplying by $\Delta$ the capacity of each element with respect to its capacity in $\bar{\mathbf{x}}$, is a feasible solution to MISC. This is so because for each set $S_i$, $\sum_{v \in S_i} \bar{x}_v \geq d_i$, and thus there must be an element $u \in S_i$ such that $\frac{d_i}{|S_i|} \leq \bar{x}_u$. Hence for this $u$, $d_i \leq |S_i| \bar{x}_u \leq \Delta \bar{x}_u$. Since this argument holds for all sets, we conclude that the solution $(\Delta\bar{x}_v)_{v \in V}$ is a feasible solution to MISC. By the linearity (in the capacities) of the objective function of MISC, we conclude that the cost of $(\Delta\bar{x}_v)_{v \in V}$ is exactly $\Delta$ times the cost of $\bar{\mathbf{x}}$ and hence at most $\Delta$ times the cost of OPT. Therefore, $(\Delta\bar{x}_v)_{v \in V}$ is a $\Delta$-approximate solution, and since (LP-MISC) can be solved in polynomial time, we established the following theorem.

**Theorem 1.** *There is a polynomial time $\Delta$-approximation algorithm for* MISC. *Therefore, there is a polynomial time* 2*-approximation algorithm for* FTC.

## 3. THE PRIMAL-DUAL ALGORITHM

We next improve the time complexity of our approximation algorithm for MISC. To do so, we extend the approximation algorithm of Hall and Hochbaum [5] for the multi-set cover problem. The multi-set cover problem (MSC) is defined, for positive integer demands $d_i$, by the following formulation:

$$\begin{array}{lll} & \min & \sum_{v\in V} w_v x_v \\ \text{(MSC)} & \text{subject to} & \sum_{v\in S_i} x_v \geq d_i \quad i = 1,\ldots m \\ & & 0 \leq x_v \leq 1 \qquad \text{integer for } v \in V. \end{array}$$

In contrast to MISC, the cover for the multi-set cover problem is formed of elements $v \in V$ each with capacity 1.

Let $A = (a_{ij})$ be the $0-1$ constraint matrix of (LP-MISC). Then LP-MISC is restated as follows.

$$\begin{array}{lll} & \min & \sum_{j=1}^{n} w_j x_j \\ \text{(LP-MISC)} & \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \geq d_i \quad i = 1,\ldots m \\ & & 0 \leq x_j \qquad \text{for } j = 1,\ldots,n. \end{array}$$

The dual to the linear programming relaxation LP-MISC is:

$$\begin{array}{lll} & \max & \sum_{i=1}^{m} d_i y_i \\ \text{(MISC-dual)} & \text{subject to} & \sum_{i=1}^{m} a_{ij} y_i \leq w_j \quad j = 1,\ldots,n \\ & & y_i \geq 0 \qquad\qquad i = 1,2,\ldots,m. \end{array}$$

In the primal-dual algorithm, we generate, like the algorithm for multi-set cover, a feasible cover along with a feasible and maximal dual solution. (For definition of "maximal" and its importance in approximating covering problems see [8].) The algorithm works, like approximation algorithms for other covering problems, by choosing at each iteration any violated constraint, or uncovered set, and covering it with the cheapest (in terms of reduced cost) available element. The distinct feature needed in this algorithm is that the selected violated constraint is the one that has the *largest* demand level among all violated constraints. We use this feature to ensure that the dual solution remains feasible throughout the algorithm.

The dual-feasible MISC algorithm:

**Step 0:** {initialize} $I = \{1,2,\ldots,m\}$, $J = V = \{1,2,\ldots,n\}$, $x_j = 0$ for all $j \in J$, $y_i = 0$ for all $i \in I$.

**Step 1:** Let $i \in I$ with $d_i = \max_{p\in I} d_p$. Let $w_k = \min\{w_j | j \in J \text{ and } a_{ij} = 1\}$. ($k$ is the minimum reduced cost column covering row $i$.) If no such minimum exists, stop - the problem is infeasible. Set $x_k = d_i$.

**Step 2:** {update of dual costs and reduced costs} Set $y_i \leftarrow y_i + w_k$. For all $j \in J$ such that $a_{ij} = 1$ set $w_j \leftarrow w_j - w_k$.

**Step 3:** {remove all sets/rows covered by element/column $k$. Their demand value can only be smaller than $d_i$} $J \leftarrow J \setminus \{k\}$. For all $i'$ such that $a_{ki'} = 1$, set $I \leftarrow I - \{i'\}$. If $I = \emptyset$ stop, else go to Step 1.

The output of the algorithm is two vectors $\mathbf{x}$ and $\mathbf{y}$ and the set of indices of elements in the cover $V \setminus J$.

From the choice of $w_k$ and the update in Step 2 all the reduced costs $w_j$ are non-negative throughout the algorithm, and the reduced costs of the elements in the cover are 0, $w_k = 0$ for all $k \in V \setminus J$.

The properties that are satisfied by the output of the algorithm $\mathbf{x}$ and $\mathbf{y}$ are as follows:

1. $\mathbf{x}$ is a feasible solution to MISC.
2. For every $j = 1,\ldots,n$ for which $x_j > 0$, its dual constraint is binding (i.e., satisfied with equality). That is because the reduced costs at the end of the algorithm are the slacks in the dual constraints.
3. The dual solution $\mathbf{y}$ is feasible for MISC-dual.

For these properties to hold it is necessary that the violated constraint is chosen to be the one with the largest demand.

We note that $A^{(1)}$, the number of nonzero entries in a $0-1$ matrix $A$, is equal to $\sum_{j=1}^{n}\sum_{i=1}^{m} a_{ij}$.

**Lemma 1.** *The time complexity of algorithm dual-feasible* MISC *is* $O(m \log n + A^{(1)})$ *which is dominated by* $O(m \log n + \Delta m)$.

**Proof.** The claim follows by noting that after sorting the demand values $d_i$ in $O(m \log n)$ time, the other operations of the algorithm take $O(1)$ time per non-zero entry of the constraint matrix, for a total of $A^{(1)}$ extra operations. Since $A^{(1)} \leq \Delta m$) the stated complexity follows. ∎

**Lemma 2.** *Algorithm dual-feasible* MISC *is a* $\Delta$*-approximation algorithm.*

**Proof.** Let $i(j)$ be the unique row for which column $j$ was selected and removed from $J$. Let the (primal) solution delivered by the algorithm be $\mathbf{x} = \mathbf{x}^H$ and the optimal solution be $\mathbf{x} = \mathbf{x}^*$. We denote the total cost of a solution $\mathbf{x}$ by $w(\mathbf{x}) = \sum_{j=1}^{n} w_j x_j$. We denote by $J'$ the subset of $J$ at the end of the algorithm corresponding to variables of value 0. We need to show that $w(\mathbf{x}^H) \leq \max_i\{\sum_j a_{ij}\} w(\mathbf{x}^*)$.

Let $\mathbf{y}$ be the dual solution delivered by the algorithm. Then, $w(\mathbf{x}^H) = \sum_{j\in V\setminus J'} w_j d_{i(j)} = \sum_{j\in V\setminus J'} (\sum_{i=1}^{m} a_{ij} y_i) d_{i(j)}$. Using the weak duality theorem we get that for any solution to the linear programming relaxation, and in particular an optimal solution $\mathbf{x}^*$ to the linear programming relaxation,

$$\begin{aligned} &\sum_{j\in V\setminus J'} \left(\sum_{i=1}^{m} a_{ij} d_{i(j)} y_i\right) \\ &\qquad \leq \max_i \left\{\sum_{j\in V\setminus J'} a_{ij}\right\} \sum_{i=1}^{m} d_{i(j)} y_i \\ &\qquad \leq \max_i \left\{\sum_{j\in V\setminus J'} a_{ij}\right\} \sum_{j=1}^{n} w_j x_j^* \\ &\qquad = \max_i \left\{\sum_{j\in V\setminus J'} a_{ij}\right\} w(\mathbf{x}^*). \end{aligned}$$

It follows that $w(\mathbf{x}^H) \leq \max_i \left\{\sum_{j\in V\setminus J'} a_{ij}\right\} w(\mathbf{x}^*) \leq \Delta \mathrm{OPT}$. ∎

With Lemma 1 and Lemma 2 we have thus established the following theorem.

**Theorem 2.** *There is an* $O(m \log n + \Delta m)$ *time* $\Delta$*-approximation algorithm for* MISC. *Therefore, there is an* $O(m \log n)$ *time* 2*-approximation algorithm for* FTC.

## REFERENCES


[1] R. Bar-Yehuda and S. Even, A linear time approximation algorithm for the weighted vertex cover algorithm, J Algorithms 2 (1981), 198–210.

[2] J. Chuzhoy and J. Naor, Covering Problems with Hard Capacities, SIAM J Comput 36 (2006), 498–515.

[3] I. Dinur and S. Safra, On the hardness of approximating minimum vertex-cover, Ann Math 162 (2005), 439–485.

[4] R. Gandhi, E. Halperin, S. Khuller, G. Kortsarz, and A. Srinivasan, An improved approximation algorithm for vertex cover with hard capacities, J Comput System Sci 72 (2006), 16–33.

[5] N.G. Hall and D.S. Hochbaum, A fast approximation algorithm for the multicovering problem, Discrete Appl Math 15 (1986), 35–40.

[6] D.S. Hochbaum, Approximation algorithms for the set covering and vertex cover problems, SIAM J Comput 11 (1982), 555–556. An extended version: Approximation algorithms for the weighted set covering and node cover problems", GSIA Working Paper No. 64-79-80, Carnegie Mellon University, Pittsburgh, PA, April 1980.

[7] D.S. Hochbaum, Efficient bounds for the stable set, vertex cover and set packing problems, Discrete Appl Math 6 (1983), 243–254.

[8] D.S. Hochbaum, Approximating covering and packing problems: set cover, vertex cover, independent set and related problems, in Approximation algorithms for NP-hard problems, D.S. Hochbaum (editor), PWS Boston, 1997, Chapter 3, pp. 94–143.

[9] G. Karakostas, A better approximation ratio for the vertex cover problem, in Proceedings of International Colloquium on Automata, Languages and Programming, Lisbon, Portugal, 2005 (ICALP 2005), pp. 1043–1050.

[10] S. Khot and O. Regev, Vertex Cover Might be Hard to Approximate to within $2 - \varepsilon$, 18th Annual IEEE Conference on Computational Complexity, Aarhus, Denmark, 2003, pp. 379–386.

[11] G. Xu, Y. Yang, and J. Xu, Linear time algorithms for approximating the facility terminal cover problem, Networks 50 (2007), 118–126.