

A Generalization of the Weighted Set Covering Problem

Jian Yang,¹ Joseph Y-T. Leung²

¹ Department of Industrial and Manufacturing Engineering, New Jersey Institute of Technology,
Newark, New Jersey 07102

² Department of Computer and Information Science, New Jersey Institute of Technology, Newark, New Jersey 07102

Received 10 October 2001; revised 31 March 2003; accepted 31 March 2003

DOI 10.1002/nav.10093

Published online 16 May 2003 in Wiley InterScience (www.interscience.wiley.com).

Abstract: We study a generalization of the weighted set covering problem where every element needs to be covered multiple times. When no set contains more than two elements, we can solve the problem in polynomial time by solving a corresponding weighted perfect b -matching problem. In general, we may use a polynomial-time greedy heuristic similar to the one for the classical weighted set covering problem studied by D.S. Johnson [Approximation algorithms for combinatorial problems, *J Comput Syst Sci* 9 (1974), 256–278], L. Lovasz [On the ratio of optimal integral and fractional covers, *Discrete Math* 13 (1975), 383–390], and V. Chvatal [A greedy heuristic for the set-covering problem, *Math Oper Res* 4(3) (1979), 233–235] to get an approximate solution for the problem. We find a worst-case bound for the heuristic similar to that for the classical problem. In addition, we introduce a general type of probability distribution for the population of the problem instances and prove that the greedy heuristic is asymptotically optimal for instances drawn from such a distribution. We also conduct computational studies to compare solutions resulting from running the heuristic and from running the commercial integer programming solver CPLEX on problem instances drawn from a more specific type of distribution. The results clearly exemplify benefits of using the greedy heuristic when problem instances are large. © 2003 Wiley Periodicals, Inc. *Naval Research Logistics* 52: 142–149, 2005.

Keywords: sets; networks/graphs; matchings; stochastic model applications

1. INTRODUCTION

We study a generalization of the classical weighted set covering problem (WSCP), the weighted multiple set covering problem (WMSCP), in which elements are required to be covered multiple times. A rigorous statement of WMSCP is as follows: There are n subsets P_1, \dots, P_n of the base set $P = \{1, 2, \dots, m\}$ such that $\cup_{j=1}^n P_j = P$ and each P_j has a positive real weight c_j . Any nonnegative-integer-valued n -tuple $x = (x_1, \dots, x_n)$ constitutes a multiple cover for P in which the number of times that element i is covered is defined to be the sum of x_j 's for those P_j 's which contain i and the total weight of the multiple cover is defined to be $\sum_{j=1}^n c_j x_j$. Given any positive-integer-valued m -tuple (t_1, \dots, t_m) , WMSCP seeks the minimum weight multiple cover for P such that every element i is covered for at least t_i times.

By defining a_{ij} to be 1 when $i \in P_j$ and 0 otherwise, we can write the problem as

$$\begin{aligned} \text{(WMSCP)} \quad & \min \sum_{j=1}^n c_j x_j \\ \text{subject to} \quad & \sum_{j=1}^n a_{ij} x_j \geq t_i \quad \forall i = 1, 2, \dots, m, \\ & x_j = 0, 1, 2, \dots \quad \forall j = 1, \dots, n. \end{aligned}$$

When all t_i 's are 1's, WMSCP becomes the well-studied WSCP [3]. Later on, for simplicity, we will always denote $\sum_{i=1}^m t_i$ by T .

Besides its theoretical significance as a generalization of the celebrated WSCP, WMSCP models a real-world problem in its own right. Consider a job shop where m type of jobs $1, \dots, m$ may be processed. Due to equipment constraints, instead of being processed individually and sequentially, jobs are simultaneously processed in operations

Correspondence to: J. Yang (yang@adm.njit.edu)

which in turn are processed sequentially. There are n types of operations P_1, \dots, P_n , where every type P_j is a subset of job types which can be simultaneously processed within this operation and it takes c_j time to complete a type P_j operation. When the demand for every job type i is t_i and we want to find the number of times x_j that every operation P_j is processed so that the demand is met with the least amount of total processing time, we need equivalently to solve a WMSCP instance.

We devote the rest of the paper to the presentation of solution methods for WMSCP and the depiction of their effectiveness. In Section 2, we show how a WMSCP instance without any set containing more than two elements can be solved in polynomial time by a simple transformation. In Section 3, we propose a polynomial-time greedy heuristic for a general WMSCP instance based on the existing greedy heuristic for WSCP. Then, we prove a worst-case upper bound for the heuristic similar to that for the existing one. In Section 4, we introduce a general type of probability distribution for the population of the WMSCP instances. We prove the asymptotic optimality of the greedy heuristic under such a distribution. In Section 5, we conduct computational comparisons between running the heuristic and running the commercial integer programming solver CPLEX to instances sampled from a more specific type of distribution. The results indicate that the greedy heuristic is both accurate and efficient when the problem size is large. In Section 6, we conclude the paper.

2. CASES SOLVABLE IN POLYNOMIAL TIME

WSCP is strongly NP -hard when some sets are allowed to have more than two elements [9]. So, as a generalization, WMSCP is strongly NP -hard as well under the same situation. Nevertheless, when no set contains more than two elements, it is well known that one can solve WSCP in polynomial time by solving a corresponding weighted perfect matching problem (WPMP). Under the same situation, we can solve WMSCP in polynomial time as well by solving a corresponding weighted perfect b -matching problem (WPBMP) [12]. Given an undirected graph $G = (V, E)$, edge weight c , and nonnegative integer vertex incidence requirement b , WPBMP seeks a nonnegative integer vector y from the following integer program:

$$(WPBMP) \quad \min \sum_{e \in E} c_e y_e$$

subject to

$$\sum_{e \text{ incident to } v} y_e = b_v \quad \forall v \in V,$$

$$y_e = 0, 1, 2, \dots \quad \forall e \in E.$$

Given a WMSCP instance, the undirected graph G for the corresponding WPBMP instance is as follows. The vertex set V contains $2m$ vertices which are labeled as $1, \dots, m$ and $1', \dots, m'$. For the edge set E , all the primed vertices are linked into a clique with each edge weighing 0. If there is a set P_j containing two elements i and k , then there is an edge linking vertex i with vertex k whose weight is c_j . And there is an edge linking every vertex i with vertex i' whose weight equals the minimum weight of any set that contains element i . As for the incidence requirement, we let $b_i = b_{i'} = t_i$.

Since every vertex i is linked with vertex i' and their incidence requirements are the same, there is guaranteed to be a perfect matching in G . Then, suppose we have found the minimum weight perfect b -matching y on G , we may transform y into a feasible solution x for the original WMSCP as follows: In the beginning, we let all x_j 's be 0. We then comb through all edges in E . If the current edge is some ik , we increase the x_j whose corresponding P_j contains exactly elements i and k by y_{ik} . If the current edge is some ii' , we increase the x_j whose corresponding P_j is one of the lightest-weighted sets that contain element i , regardless of whether it has two elements or one element, by $y_{ii'}$. If the current edge is some $i'k'$, we do nothing.

We can prove

THEOREM 1: An optimal perfect b -matching y for the WPBMP instance corresponding to the original WMSCP instance is transformed into an optimal multiple cover x for the original WMSCP instance.

We leave the proof to the Appendix since it is very standard.

Similar to WBMP, we can solve WPBMP by solving an equal-sized weighted maximum b -matching problem (WMBMP) [12]. Anstee [1], Cunningham and Marsh [4], and Gabow [6] have all given polynomial-time algorithms for WMBMP. In particular, the time complexity of Anstee's algorithm does not depend on the sizes of the input edge weights and targeted incidence levels: $O(|E||V|^2 \ln(|V|) + |V|^3 \ln^2(|V|))$. Given a WMSCP instance with m elements and n sets ($n \leq m(m + 1)/2$), we can construct the corresponding WPBMP instance with $2m$ vertices and $n + m(m + 1)/2$ edges in $O(m^2)$ time, solve the WPBMP instance in $O(m^4 \ln(m))$ time by solving a corresponding WMBMP instance, and transform the solution for the WPBMP instance into the solution for the original WMSCP instance in $O(m^2)$. Hence, the total time complexity for our solution method toward the special WMSCP is $O(m^4 \ln(m))$.

3. GENERAL CASES AND A GREEDY HEURISTIC

For the unweighted set covering problem (SCP), a natural heuristic is a greedy one which iteratively selects a set with the maximum number of uncovered elements [8, 10]. Chvatal [3] extended this to a heuristic suitable for WSCP in which sets are still iteratively selected and, in every iteration, the set with the minimum average weight per uncovered element is selected. The widely known worst-case upper bound of the performance ratio for this heuristic is $H(d) = \sum_{k=1}^d 1/k$, where d is the cardinality of the largest set. If we consider the instances with the same m all together, this upper bound is about $\ln(m)$. On the other hand, the bound is almost the best we could hope for both SCP and WSCP due to the results by Lund and Yannakakis [11] and Feige [5]. Also, Slavik [13] proved that the worst-case performance ratio is exactly $\ln(m) - \ln(\ln(m)) + \Theta(1)$ for SCP.

Here, for WMSCP, we propose a similar greedy heuristic (GH) and prove a similar upper bound. The description of GH is as follows:

- Step 0. Let $x_j^* = 0$ for $j = 1, \dots, n$.
- Step 1. If $t_i = 0$ for every i , then stop: $x^* = (x_1^*, \dots, x_n^*)$ is the final solution.
- Otherwise, let $k = \operatorname{argmax}_{j=1}^n |P_j|/c_j$, an index j that achieves the largest $|P_j|/c_j$ among all possible choices of j in $\{1, 2, \dots, n\}$, and proceed to Step 2.
- Step 2. Let $t^* = \min_{i \in P_k} t_i$. Let $x_k^* = x_k^* + t^*$. For every $i \in P_k$, let $t_i = t_i - t^*$, and if $t_i = 0$, let $P_j = P_j \setminus \{i\}$ for every j . Go back to Step 1.

GH runs in iterations. Since every iteration completely covers at least one element, the number of iterations s does not exceed m . In every iteration, it takes $O(n)$ time to choose the set with the maximum remaining-size/weight ratio and $O(m)$ time to calculate the step size t^* and to see what updatings of sets are required. There are exactly m times that all n sets need to be updated and the updating on every set takes $O(m)$ time. So, GH runs in $O(mn + m^2)$ time.

Let x^* stand for the solution given by GH and x be any feasible solution of WMSCP; we have

THEOREM 2:

$$\frac{\sum_{j=1}^n c_j x_j^*}{\sum_{j=1}^n c_j x_j} \leq H(d),$$

where d is the cardinality of the largest set among P_1, \dots, P_n .

PROOF: The proof is mainly built upon the work by Chvatal [3]. We follow his notation as much as possible throughout this paper. Denote every P_j at the beginning of iteration r by P_j^r and the index of the set that is chosen in the iteration by (r) . Assuming $c_{(0)}/|P_{(0)}^0| = 0$, for $r = 1, \dots, s, j = 1, \dots, n$, we have

$$\frac{c_{(r-1)}}{|P_{(r-1)}^{r-1}|} \leq \frac{c_{(r)}}{|P_{(r)}^{r-1}|} \leq \frac{c_j}{|P_j^r|}.$$

For $k = 1, \dots, t_i$, we can attribute a weight y_i^k to element i 's k th coverage. We can easily obtain that

$$y_i^1 \leq y_i^2 \leq \dots \leq y_i^{t_i}$$

and

$$\sum_{j=1}^n c_j x_j^* = \sum_{i=1}^m \sum_{k=1}^{t_i} y_i^k.$$

For the feasible solution x , we have

$$1 \leq \frac{1}{t_i} \sum_{j=1}^n a_{ij} x_j.$$

The following is the most important observation toward the proof: Assuming that $P_j^{s+1} = \emptyset$, for an arbitrary element i in an arbitrary set P_j , the element is completely covered (covered for the last time) in an arbitrary iteration r if and only if $i \in P_j^r \setminus P_j^{r+1}$. From this, we can derive the key identity

$$\sum_{i=1}^m y_i^{t_i} a_{ij} = \sum_{r=1}^s \frac{c_{(r)}}{|P_{(r)}^{r-1}|} (|P_j^r| - |P_j^{r+1}|).$$

Also, an important inequality is in order:

$$\sum_{r=1}^s \frac{1}{|P_j^r|} (|P_j^r| - |P_j^{r+1}|) \leq \sum_{r=1}^s \sum_{z=|P_j^{r+1}|+1}^{|P_j^r|} \frac{1}{z} = H(|P_j|) \leq H(d).$$

Now, we can proceed with the proof without much more effort:

$$\begin{aligned} \sum_{j=1}^n c_j x_j^* &= \sum_{i=1}^m \sum_{k=1}^{t_i} y_i^k \leq \sum_{i=1}^m \left(\frac{1}{t_i} \sum_{k=1}^{t_i} y_i^k \sum_{j=1}^n a_{ij} x_j \right) \\ &\leq \sum_{j=1}^n \left(\sum_{i=1}^m y_i^{t_i} a_{ij} \right) x_j = \sum_{j=1}^n \left(\sum_{r=1}^s \frac{c_{(r)}}{|P_{(r)}^{r-1}|} (|P_j^r| - |P_j^{r+1}|) \right) x_j \\ &\leq \sum_{j=1}^n \left(\sum_{s=1}^r \frac{1}{|P_j^s|} (|P_j^s| - |P_j^{s+1}|) \right) c_j x_j \leq H(d) \sum_{j=1}^n c_j x_j. \quad \square \end{aligned}$$

This worst-case bound is tight among all instances with $t_1 = \dots = t_m = 1$ [3], so we can say that the bound is also tight among all WMSCP instances. Nevertheless, when we consider only all instances with a specific m -tuple (t_1, \dots, t_m) which is not all-one, a tight worst-case upper bound might have to depend on the t_i 's. So far, we have not been able to find this tight bound, nor have we proved that the $H(d)$ bound is indeed tight for all instances with a given m -tuple. For a given (t_1, \dots, t_m) with $t_1 \leq \dots \leq t_m$ (without loss of generality), the highest lower bound that we have found is $\sum_{i=1}^m t_i/(t_1 \times i)$. The following is the instance where this bound is achieved: There are $2m - 1$ sets. For $i = 1, \dots, m$, $P_i = \{m + 1 - i\}$ and $c_i = 1/(m + 1 - i)$, and for $i = 1, \dots, m - 1$, $P_{m+i} = \{1, 2, \dots, i + 1\}$ and $c_{m+i} = 1$. Applying GH to this problem, we could get $x_i^* = t_{m+1-i}$ for $i = 1, \dots, m$ and $x_{m+i}^* = 0$ for $i = 1, \dots, m - 1$. On the other hand, the optimal solution is $x_i = 0$ for $i = 1, \dots, m - 1$, $x_{m+i} = t_{i+1} - t_{i+2}$ for $i = 0, \dots, m - 2$, and $x_{2m-1} = t_m$. We have

$$\sum_{j=1}^{2m-1} c_j x_j^* = \sum_{i=1}^m \frac{t_i}{i} \quad \text{and} \quad \sum_{j=1}^{2m-1} c_j x_j = t_1.$$

4. ASYMPTOTIC BEHAVIOR OF THE GREEDY HEURISTIC

In this section, we introduce a type of probability distribution for the problem instance population of WMSCP and show that GH is asymptotically optimal under such a probability distribution. We describe the type of distribution by describing the way under which a sample instance is drawn for given m , n , and t_1, \dots, t_m . First, every a_{ij} is an independent Bernoulli random variable with a certain $p_i \in (0, 1)$ as the chance of a_{ij} being 1. If under these a_{ij} 's every element i is contained in some set P_j , the drawing of the a_{ij} 's is completed. Otherwise, we assign the same independent Bernoulli random variables to a_{ij} 's once again. We keep going on with the rounds till every element is contained in some set. Apparently, the result of every round is independent of those of other rounds and the chance of success for every round is $\prod_{i=1}^m [1 - (1 - p_i)^n]$. This chance tends to $1 - \sum_{i=1}^m (1 - p_i)^n$ which tends to 1 as n approaches ∞ . With the possible abandonment of any single round, the final a_{ij} 's are no longer independent of each other. However, given that certain elements are contained in certain sets, any joint distribution of the a_{ij} 's that involves only these certain elements and that does not involve these certain sets remains the same as when the a_{ij} 's are generated in one single round. Then, after the a_{ij} 's have been decided, we let the unit weights $c_j/|P_j|$ of all the sets be independent identically distributed random variables with a certain distribution $F(\cdot)$ which satisfies

$$\underline{u} = \inf_{F(x) > 0} x \geq 0$$

and

$$\begin{aligned} f &= \lim_{x \rightarrow \underline{u}^+} \left[\inf_{y \in [\underline{u}, x]} (F(x) - F(y^-))/(x - y) \right] \\ &= \sup_{n=1}^{\infty} \inf_{\underline{u} \leq y < x < \underline{u} + 1/n} (F(x) - F(y^-))/(x - y) > 0. \end{aligned}$$

Note that, when coincidentally two equal sets with different weights are generated, the heavier-weighted set can be assumed to be nonexistent.

For convenience, we use p_M to denote $\max_{i=1}^m p_i$. We know that $1 - p_M > 0$. Also let $\Delta u > 0$ satisfy that, for any $\underline{u} \leq y < x \leq \underline{u} + \Delta u$,

$$\frac{F(x) - F(y^-)}{x - y} \geq \frac{f}{2}.$$

According to the above, we know that this Δu must exist. Let $(A[n], c[n])$ be a random instance generated from the above distribution with given m , t_i 's, p_i 's, and n , where $A[n]$ stands for the incidence matrix and $c[n]$ stands for the weight vector. Denote the total weight produced by running GH on $(A[n], c[n])$ by $GH(n)$ and the total weight produced by running OPT on $(A[n], c[n])$ by $OPT(n)$.

Since \underline{u} is almost surely the lowest weight associated with a unit demand, we have

$$OPT(n) \geq T\underline{u} \quad \text{a.s.}$$

On the other hand, we have the following result regarding $GH(n)$:

THEOREM 3: For every $\epsilon \in (0, \Delta u)$,

$$\lim_{n \rightarrow \infty} Pr[GH(n) \leq T\underline{u} + \epsilon] = 1.$$

PROOF: From the range of ϵ , we have

$$Pr \left[\frac{c_j}{|P_j|} \leq \underline{u} + \epsilon \right] = F(\underline{u} + \epsilon) - F(\underline{u}^-) \geq \frac{f\epsilon}{2} > 0$$

and therefore

$$\lim_{n \rightarrow \infty} Pr \left[\frac{c_{(1)}}{|P_{(1)}|} \leq \underline{u} + \epsilon \right] \geq \lim_{n \rightarrow \infty} \left[1 - \left(1 - \frac{f\epsilon}{2} \right)^n \right] = 1.$$

Also, for any $r \geq 2$, for any realization of $(1), \dots, (r - 1)$ such that all demands of the elements are still not fully met by the selection of the sets $P_{(1)}, \dots, P_{(r-1)}$ and that $c_{(r-1)}/|P_{(r-1)}^{r-1}|$ is still smaller than $\underline{u} + \Delta u$, for any $\epsilon \in (0,$

$\Delta u - c_{(r-1)}/|P_{(r-1)}^{r-1}| + \underline{u}$, and for any j other than $(1), \dots, (r-1)$,

$$\begin{aligned} Pr \left[P_j^r = P_j \quad \text{and} \quad \frac{c_{(r-1)}}{|P_{(r-1)}^{r-1}|} \leq \frac{c_j}{|P_j^r|} \leq \frac{c_{(r-1)}}{|P_{(r-1)}^{r-1}|} + \epsilon \right] \\ \times (1), \dots, (r-1) \\ = Pr[P_j^r = P_j | (1), \dots, (r-1)] Pr \left[\frac{c_{(r-1)}}{|P_{(r-1)}^{r-1}|} \leq \frac{c_j}{|P_j^r|} \right] \\ \leq \frac{c_{(r-1)}}{|P_{(r-1)}^{r-1}|} + \epsilon \left| (1), \dots, (r-1) \right| \geq (1 - p_M)^{m-1} \\ \times \frac{F(c_{(r-1)}/|P_{(r-1)}^{r-1}| + \epsilon) - F((c_{(r-1)}/|P_{(r-1)}^{r-1}|)^-)}{1 - F((c_{(r-1)}/|P_{(r-1)}^{r-1}|)^-)} \\ \geq \frac{(1 - p_M)^{m-1} \underline{f} \epsilon}{2} > 0. \end{aligned}$$

In the above derivation, we have used the facts that the selection of unit weights and selection of a_{ij} 's are independent, that

$$[P_j^r = P_j | (1), \dots, (r-1)] = [a_{ij} = 0$$

for all elements i whose demands are fully met by

$$P_{(1)}, \dots, P_{(r-1)} | (1), \dots, (r-1)],$$

that the note in the last paragraph on the independence of a_{ij} 's applies here, and that the number of fully covered elements is at most $m - 1$. From the above result, we have, furthermore, that

$$\begin{aligned} \lim_{n \rightarrow \infty} Pr \left[P_{(r)}^r = P_r \quad \text{and} \quad \frac{c_{(r-1)}}{|P_{(r-1)}^{r-1}|} \leq \frac{c_r}{|P_{(r)}^r|} \leq \frac{c_{(r-1)}}{|P_{(r-1)}^{r-1}|} \right. \\ \left. + \epsilon \right| (1), \dots, (r-1) \\ \geq \lim_{n \rightarrow \infty} \left[1 - \left(1 - \frac{(1 - p_M)^{m-1} \underline{f} \epsilon}{2} \right)^n \right] = 1. \end{aligned}$$

It is easy to find a conservative rate of convergence that has nothing to do with the specifications of $(1), \dots, (r-1)$.

Recall that s denotes the number of iterations used in GH. For $r = 1, \dots, m$, if $Pr[s \geq r]$ is positive, then the convergences in probability in the last paragraph should also be true when the probabilities are replaced with probabilities conditioned on $s \geq r$. Also, we have

$$GH(n) = t_1^* c_{(1)} + \dots + t_s^* c_{(s)},$$

$$t_1^* |P_{(1)}^1| + \dots + t_s^* |P_{(s)}^s| = T,$$

$|P_{(r)}^r| \leq m$ for any r , and $s \leq T$. Therefore, the combination of

$$\frac{c_{(1)}}{|P_{(1)}^1|} \leq \underline{u} + \frac{\epsilon}{mT},$$

and, for $r = 2, \dots, s$,

$$P_{(r)}^r = P_r \quad \text{and} \quad \frac{c_r}{|P_{(r)}^r|} \leq \frac{c_{(r-1)}}{|P_{(r-1)}^{r-1}|} + \frac{\epsilon}{mT}$$

will imply that

$$\begin{aligned} GH(n) &= t_1^* |P_{(1)}^1| \times \frac{c_1}{|P_{(1)}^1|} + \dots + t_s^* |P_{(s)}^s| \times \frac{c_s}{|P_{(s)}^s|} \\ &\leq t_1^* |P_{(1)}^1| \left(\underline{u} + \frac{\epsilon}{mT} \right) + \dots + t_s^* |P_{(s)}^s| \left(\underline{u} + s \frac{\epsilon}{mT} \right) \\ &\leq T \underline{u} + \epsilon. \end{aligned}$$

In the above, note that $P_{(1)} = P_{(1)}^1$ is always true.

Hence, under a particular s with a positive probability, we have

$$\begin{aligned} \lim_{n \rightarrow \infty} Pr [GH(n) \leq T \underline{u} + \epsilon] &\geq \lim_{n \rightarrow \infty} Pr \left[\frac{c_{(1)}}{|P_{(1)}^1|} \leq \underline{u} + \frac{\epsilon}{mT} \right] \\ &\times Pr \left[P_{(2)}^2 = P_{(2)} \quad \text{and} \quad \frac{c_{(1)}}{|P_{(1)}^1|} \leq \frac{c_{(2)}}{|P_{(2)}^2|} = P_{(2)} \leq \frac{c_{(1)}}{|P_{(1)}^1|} \right. \\ &\quad \left. + \frac{\epsilon}{mT} \right| (1) \times \dots \times Pr \left[P_{(s)}^s = P_{(s)} \quad \text{and} \right. \\ &\quad \left. \frac{c_{(s-1)}}{|P_{(s-1)}^{s-1}|} \leq \frac{c_{(s)}}{|P_{(s)}^s|} \leq \frac{c_{(s-1)}}{|P_{(s-1)}^{s-1}|} + \frac{\epsilon}{mT} \right| (1), \dots, (s-1) \\ &\geq \lim_{n \rightarrow \infty} Pr \left[\frac{c_{(1)}}{|P_{(1)}^1|} \leq \underline{u} + \frac{\epsilon}{mT} \right] \times \lim_{n \rightarrow \infty} \inf_{(1)} Pr \left[P_{(2)}^2 = P_{(2)} \right. \\ &\quad \left. \text{and} \quad \frac{c_{(1)}}{|P_{(1)}^1|} \leq \frac{c_{(2)}}{|P_{(2)}^2|} \leq \frac{c_{(1)}}{|P_{(1)}^1|} + \frac{\epsilon}{mT} \right| (1) \times \dots \\ &\times \lim_{n \rightarrow \infty} \inf_{(1), \dots, (s-1)} Pr \left[P_{(s)}^s = P_{(s)} \quad \text{and} \quad \frac{c_{(s-1)}}{|P_{(s-1)}^{s-1}|} \leq \frac{c_{(s)}}{|P_{(s)}^s|} \right. \\ &\quad \left. \leq \frac{c_{(s-1)}}{|P_{(s-1)}^{s-1}|} + \frac{\epsilon}{mT} \right| (1), \dots, (s-1) = 1. \quad \square \end{aligned}$$

Table 1. GH vs. OPT (I): $t = 1$ and $p = 0.1$.

n	GH	$OPT[10.0]$	GH/OPT	T_{GH}	T_{OPT}
100	15.32	13.85	1.11	0.000	0.006
200	12.77	11.88	1.08	0.000	0.010
500	11.24	10.75	1.05	0.001	0.018
1000	10.59	10.36	1.02	0.002	0.034
2000	10.33	10.19	1.01	0.005	0.069
5000	10.13	10.07	1.01	0.015	0.215
10,000	10.06	10.04	1.00	0.038	0.440
20,000	10.03	10.02	1.00	0.080	0.473

Table 3. GH vs. OPT (I): $t = 10$ and $p = 0.1$.

n	GH	$OPT[10.0]$	GH/OPT	T_{GH}	T_{OPT}
100	88.54	80.90	1.09	0.000	0.004
200	71.56	67.95	1.07	0.001	0.007
500	63.68	61.05	1.04	0.001	0.013
1000	59.26	57.91	1.02	0.003	0.027
2000	57.78	57.13	1.01	0.005	0.059
5000	56.15	55.89	1.00	0.020	0.192
10,000	55.42	55.28	1.00	0.054	0.409
20,000	54.83	54.76	1.00	0.115	0.398

A very conservative estimate on the convergence rate of the above convergence in probability result is also in order. For any $\epsilon \in (0, \Delta u)$ and any positive δ that are small enough, let

$$N(\epsilon, \delta) = T + \frac{\log(T/\delta)}{\log(2mT/(2mT - (1 - p_M)^{m-1}\underline{f}\epsilon))}.$$

Then, for any $n > N(\epsilon, \delta)$, we have

$$1 - \delta \leq Pr[T\underline{u} \leq GH(n) \leq T\underline{u} + \epsilon] \leq 1.$$

Clearly, $N(\epsilon, \delta)$ is an increasing function of $m, p_M, 1/\underline{f}$, and T . Roughly speaking, the rate of convergence decreases when either one of $m, p, 1/\underline{f}$, or T increases.

Furthermore, when $\underline{u} > 0$ and $\bar{u} = \sup_{F(x) < 1} x < \infty$, we apparently have

$$\lim_{n \rightarrow \infty} E \left[\frac{GH(n)}{OPT(n)} \right] = 1.$$

When t_i 's are random so that T has a finite bound \bar{T} and when the selection of them is independent of the selection of the sets and the unit set weights, we shall have that $E[GH(n)]$ converges to $E[T]\underline{u}$ and $E[GH(n)/OPT(n)]$ converges to 1 with the rates of convergence being roughly decreasing in $m, p_M, 1/\underline{f}$, and \bar{T} .

Table 2. GH vs. OPT (I): $t = 1$ and $p = 0.5$.

n	GH	$OPT[10.0]$	GH/OPT	T_{GH}	T_{OPT}
100	17.68	15.47	1.15	0.000	0.079
200	16.36	14.04	1.17	0.000	0.137
500	14.98	12.81	1.17	0.001	0.497
1000	14.27	12.26	1.17	0.003	1.508
2000	13.76	11.68	1.18	0.006	4.505
5000	13.27	11.84	1.12	0.017	10.002
10,000	12.64	11.83	1.07	0.037	10.500
20,000	12.33	11.99	1.03	0.075	11.114

Table 4. GH vs. OPT (I): $t = 10$ and $p = 0.5$.

n	GH	$OPT[10.0]$	GH/OPT	T_{GH}	T_{OPT}
100	101.10	86.96	1.16	0.001	0.019
200	92.70	79.65	1.16	0.001	0.054
500	83.33	70.21	1.19	0.002	0.181
1000	78.79	66.18	1.19	0.003	0.601
2000	74.80	62.80	1.19	0.010	2.250
5000	71.25	60.35	1.18	0.038	7.000
10,000	67.70	58.48	1.16	0.082	9.975
20,000	65.44	57.67	1.14	0.165	10.938

5. COMPUTATIONAL RESULTS

In this section, we conduct a computational study under a more specific type of probability distribution to compare GH with OPT . The only parameters that identify a distribution here are $p \in (0, 1)$, $v \in (0, 1)$, and $t \in \{1, 2, \dots\}$, where p replaces the p_i 's in the previous more general distribution, the unit weight of a set is uniformly distributed in $[1 - v, 1 + v]$, and every t_i is uniformly distributed in $\{1, 2, \dots, t\}$. We shall have $p_M = p$, $\underline{u} = 1 - v$, $\underline{f} = 1/(2v)$, $E[T] = m(t + 1)/2$, and $\bar{T} = mt$. When fixing m, p, v , and t , and letting n tend to ∞ , $GH(n)$ should converge to $m(t + 1)(1 - v)/2$ with a rate that is roughly decreasing in m, p, v , and t .

We run CPLEX 7.0 on the mathematical formulation to get OPT . CPLEX is a commercial integer programming solver distributed by ILOG [7]. As is the same in reality, we impose limitation on how much time can be spent by CPLEX on every instance (In the range of our study, GH always ends at most in seconds thus requires no time limitations). For a given instance, we use GH to denote the solution found by GH and $OPT[\tau]$ to denote the best feasible solution found by CPLEX in τ seconds, and use GH/OPT to denote the performance ratio of the two results. We also record the real amount of time elapsed while running each method: T_{GH} for GH and T_{OPT} for OPT . For every fixed set of parameters (m, n, p, v, t) , we independently generate 100 WMSCP instances following the probabilistic model in the last section, run GH and OPT on these instances, and present the average results.

In Tables 1, 2, 3, and 4, we present the results for $m =$

Table 5. GH vs. OPT (II): $m = 100$.

n	GH	$OPT[60.0]$	GH/OPT	T_{GH}	T_{OPT}
100	774.71	672.92	1.15	0.002	0.558
200	616.66	535.61	1.15	0.003	13.103
500	496.51	428.79	1.16	0.007	59.093
1000	470.00	383.63	1.15	0.013	60.046
2000	407.65	353.65	1.15	0.030	60.084
5000	363.68	340.18	1.13	0.133	60.201
10,000	350.36	315.16	1.11	0.309	60.541

20, $v = 0.5$, two different t 's, two different p 's, and various n 's. Table 1 contains the results for $t = 1$ and $p = 0.1$, Table 2 contains the results for $t = 1$ and $p = 0.5$, Table 3 contains the results for $t = 10$ and $p = 0.1$, and Table 4 contains the results for $t = 10$ and $p = 0.5$.

From the four tables, we see that $E[GH(n)]$ indeed converges to $m(t + 1)(1 - v)/2$. For Tables 1 and 2, this constant is 10.00, and, for Tables 3 and 4, this constant is 55.00. Also, it is very obvious that the rate of convergence decreases in the density p and the maximum demand t per element.

In Table 5, we present the results on the much larger problems where $m = 100$, $p = 0.1$, $v = 0.5$, $t = 10$, and n varies.

In Table 5, GH seems to converge to the targeted 275.00, not withstanding with a much lower rate than that reflected by Table 3 where m is smaller. The table also tells us that, for problems with hundreds of elements and thousands of sets, OPT relying on CPLEX does not provide a reasonably good solution in a reasonable amount of time most of the time while GH does in a fraction of time taken by the former method. The advantage of GH over OPT is even more apparent when problem instances get larger.

In the following four diagrams, we present $GH/[m(t + 1)(1 - v)/2]$ and $OPT[300.0]/[m(t + 1)(1 - v)/2]$ for

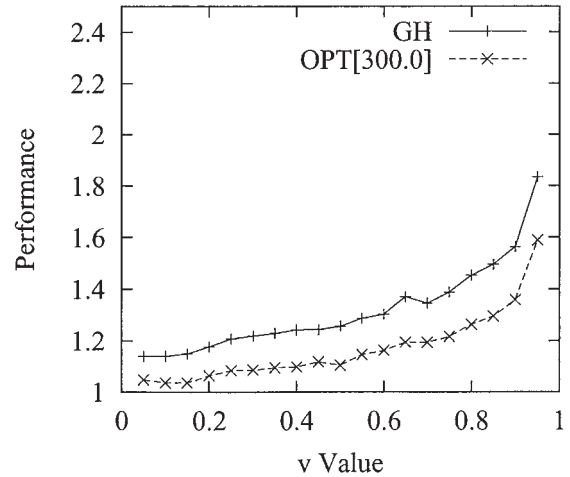


Figure 2. GH vs. $OPT[300.0]$ over varying v 's.

problems with $n = 10,000$, varying m 's, varying v 's, varying p 's, and varying t 's. We let $m = 100$, $v = 0.5$, $p = 0.1$, and $t = 10$ when any one of them is not varying. Due to time constraints, every point in the diagrams is the average of results from 10 independent runs instead of from 100 independent runs.

From the figures, we can make the following conclusions. When m , v , or t increases, the advantage of GH over $OPT[\tau]$ becomes ever more apparent. When m , v , or p increases, convergences to the asymptotic behavior of both GH and OPT slow down. On the other hand, $OPT[\tau]$ can still outperform GH when p increases to a certain degree. In addition, at $t = 1$, i.e., when the problem reduces to WSCP, both GH and $OPT[\tau]$ can converge to the asymptotic results very fast in terms of n , while for other t 's, even for $t = 2$, the convergences are much slower.

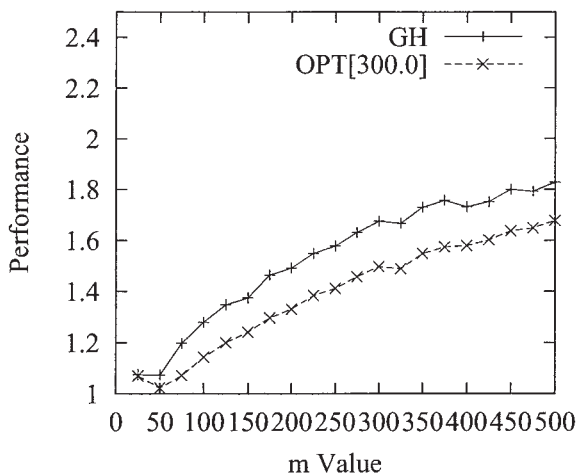


Figure 1. GH vs. $OPT[300.0]$ over varying m 's.

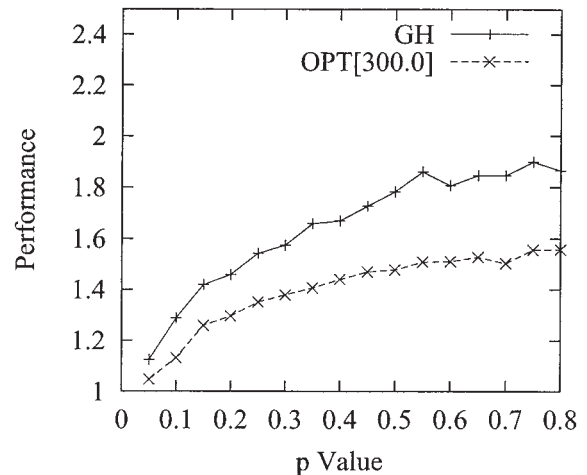


Figure 3. GH vs. $OPT[300.0]$ over varying p 's.

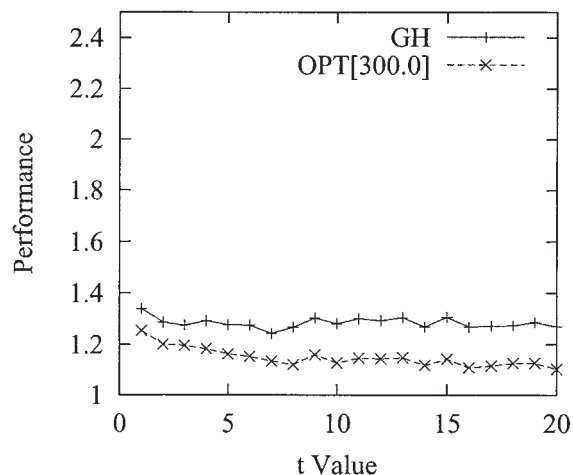


Figure 4. GH vs. OPT[300.0] over varying t 's.

6. CONCLUDING REMARKS

We have proposed WMSCP as a generalization of the much-studied WSCP and SCP. We have then shown that a special case of the problem can be solved in polynomial time and proposed a polynomial-time greedy heuristic GH for tackling the general case. We have also established a worst-case performance upper bound for GH with potential for improvement. And, under a certain type of probabilistic distribution, we have shown that GH gives asymptotically optimal answer for WMSCP. Our computational study has confirmed the benefit of using GH.

Although existing probabilistic analyses of combinatorial optimization problems are abundant, we have not seen any such study done directly towards WSCP. For SCP, a framework to study the average-case behavior of certain heuristics was proposed by Blot et al. [2]. However, the greedy heuristic proposed by Johnson [8] and Lovasz [10], the ancestor of GH, was not studied there. Hence, our asymptotic result, though not surprising at all, is the first one in its own category.

APPENDIX

PROOF OF THEOREM 1: First, the x resulting from the transformation is feasible for the original WMSCP instance: The solution y ensures that every vertex i has exactly t_i number of incidences and according to the transformation, every such incidence is one unit of coverage for element i . In addition, the total WPBMP weight of y equals the total WMSCP weight of x . Then, we need only to show that the above x is optimal for WMSCP. We achieve this by showing that the optimal x^* for WMSCP can be transformed back into a feasible y^* for WPBMP with the same weight. If so, x has to be optimal for WMSCP, for otherwise y will not be optimal for WPBMP, which is a contradiction.

Given the x^* , we start with y^* being 0 and comb through all the j 's to build up y^* . During the process, for every vertex i , let its current incidence level resulting from the current y^* be z_i^* . In the beginning, z_i^* is 0. First, for every j for which set P_j is a two-element set $\{i, k\}$ that is not the

lightest-weighted among either all sets containing element i or all sets containing element k , increase y_{ik}^* by x_j^* . Then, for every j for which set P_j is a one-element set $\{i\}$, increase y_{ii}^* by x_j^* . And, for every j for which set P_j is a two-element set $\{i, k\}$ that is the lightest-weighted among all sets containing element i while not among all sets containing element k , increase y_{ik}^* by $\delta = \min\{x_j^*, t_k - z_k^*\}$ and y_{ii}^* by $(x_j^* - \delta)^+$. Finally, for every j for which set P_j is a two-element set $\{i, k\}$ that is the lightest-weighted among all sets containing element i or k , suppose $t_i - z_i^* \geq t_k - z_k^*$; then increase y_{ik}^* by $\delta = \min\{x_j^*, t_k - z_k^*\}$ and y_{ii}^* by $(x_j^* - \delta)^+$.

Now we claim that every $z_i^* = t_i$ and every $z_{i'}^* \leq t_{i'}$. Thus we can increase the $y_{i'k'}^*$'s appropriately to bring the $z_{i'}^*$'s up to $t_{i'}$'s since the total unfulfilled incidence level is an even number and all the i' 's are linked into a clique. In this way, y^* is a feasible solution for WPBMP and its weight is the same as that of x^* . To briefly prove the last claim, we only need to note the following. From the feasibility of x^* for WMSCP, we have in the end that $z_i^* \geq t_i$ for every i . But from the optimality of x^* , no z_i^* should exceed t_i after only considering the first two types of sets. When considering the last two types of sets, we have ensured that $z_i^* \leq t_i$ for every i . That $z_{i'}^* \leq t_{i'}$ is again due to the optimality of x^* . \square

ACKNOWLEDGMENTS

Research by Jian Yang is supported, in part, by New Jersey Institute of Technology under Grant No. 4-21830 and the National Center for Transportation and Industrial Productivity under Grant No. 9-92518.

REFERENCES

- [1] R.P. Anstee, A polynomial algorithm for b -matching: An alternative approach, *Inform Process Lett* 24 (1987), 153–157.
- [2] J. Blot, W.F. De La Vega, V.T. Paschos, and R. Saad, Average case analysis of greedy algorithms for optimisation problems on set systems, *Theoret Comput Sci* 147 (1995), 267–298.
- [3] V. Chvatal, A greedy heuristic for the set-covering problem, *Math Oper Res* 4:3 (1979), 233–235.
- [4] W.H. Cunningham and A.B. Marsh, A primal algorithm for optimum matching, *Math Prog Study* 8 (1978), 50–72.
- [5] U. Feige, A threshold of $\ln n$ for approximating set cover, *J ACM* 45(4) (1998), 634–652.
- [6] H.N. Gabow, An efficient reduction technique for constrained subgraph and bidirected network flow problems, *Conf Proc Annu ACM Symp Theory Comput* 15 (1983), 448–456.
- [7] ILOG CPLEX 7.0-User's Manual, ILOG, Mountain View, California, 2000.
- [8] D.S. Johnson, Approximation algorithms for combinatorial problems, *J Comput System Sci* 9 (1974), 256–278.
- [9] R.M. Karp, "Reducibility among combinatorial problems," *Complexity of computer computations*, R.E. Miller and J.W. Thatcher (Editors), Plenum Press, New York, 1972 pp. 85–104.
- [10] L. Lovasz, On the ratio of optimal integral and fractional covers, *Discrete Math* 13 (1975), 383–390.
- [11] C. Lund and M. Yannakakis, C. Lund, and M. Yannakakis, On the hardness of approximating minimization problems, *J ACM* 41(5) (1994), 960–981.
- [12] G.L. Nemhauser and L.A. Wolsey, *Integer and combinatorial optimization*, Wiley Interscience, New York, 1988.
- [13] P. Slavik, A tight analysis of the greedy algorithm for set covering, *Conf Proc Annu ACM Symp Theory Comput* 28 (1996), 435–441.