JOURNAL OF ALGORITHMS 2, 198–203 (1981)

# A Linear-Time Approximation Algorithm for the Weighted Vertex Cover Problem

R. BAR-YEHUDA AND S. EVEN*

*Department of Computer Science, Technion–Israel Institute of Technology, Haifa, Israel*

Received January 21, 1981

A linear time approximation algorithm for the weighted set-covering problem is presented. For the special case of the weighted vertex cover problem it produces a solution of weight which is at most twice the weight of an optimal solution.

## INTRODUCTION

The weighted set-covering problem can be stated as follows. There is a finite family of a finite sets: $S_1, S_2, \ldots, S_n$, and each set $S_i$ has a given weight $w_i$, which is a real nonnegative number. Let $U$ be $\cup_{i=1}^{n} S_i$ (the universal set). One wants to find a subset of the family specified by $I \subseteq \{1, 2, \ldots, n\}$ such that

$$\bigcup_{i \in I} S_i = U, \tag{1}$$

for which the total weight

$$\sum_{i \in I} w_i \tag{2}$$

is minimum. This problem is known to be NP-hard, even if all the weights are equal to 1 (see, for example, [1], [2], or [3]).

In view of this fact one is naturally led to search for polynomial-time approximation algorithms. The best known result is by Chvátal [4]. His algorithm can be implemented easily in time $O(m \log n)$, where $m = \sum_{i=1}^{n} |S_i|$, and the weight of its result, $W$, satisfies

$$W \leq W_{\text{opt}} \cdot O(\log d), \tag{3}$$

*Parts of this research were done while S. Even visited the Computer Science Division, EECS, University of California, supported by NSF Contract MCS 7915763, and while he visited the Mathematical Institute of the Technical University of Denmark, Lyngby, Denmark.

where $W_{opt}$ is the weight of an optimal solution and $d = \text{Max}_i|S_i|$.

Lately, another polynomial-time approximation algorithm has been proposed by Hochbaum [5]. She uses a solution of a linear programming problem, and therefore, the proof that her algorithm is polynomial relies on the Russian algorithm for solving linear programming in polynomial time (see, for example, [6]). The weight of the solution of Hochbaum's algorithm, $W$, satisfies the condition

$$W \leq W_{opt} \cdot f, \tag{4}$$

where $f$ is the largest number of sets which contain a certain element; i.e., let $e_1, e_2, \ldots, e_t$ be the elements in $U$. Define

$$F(j) = \{i \mid e_j \in S_i\};$$

then

$$f = \text{Max}|F(j)|.$$

This result is usually inferior to Chvátal's result, but in the case of the vertex cover, it is most attractive.

The weighted vertex cover problem is as follows. One is given a finite undirected graph $G(V,E)$, and each vertex $v$ is assigned a nonnegative weight $\omega(v)$. The goal is to find a subset of vertices, $S$, such that for every edge, at least one of its endpoints is in $S$, and $S$ has the minimum weight $W(S) = \Sigma_{v \in S}\omega(v)$, of all such subsets.

This vertex cover (VC) problem is also NP-hard, even if all the vertex weights are equal to 1 (see [1], [2] or [3]).

There are many simple linear time (in $|V| + |E|$) approximation algorithms, for finding a solution of the unweighted VC problem, which is only at most twice as large as the minimum solution. The first recorded such solution is due to Gavril (see [2, p. 134]). It goes as follows. Find a maximal (but not necessarily maximum) matching in the graph, i.e., a maximal subset of edges, $M$, such that no two edges in $M$ have an endpoint in common. Let $S$ be the set of all endpoints of edges in $M$. Clearly, since $M$ is maximal, there cannot be an edge in $E - M$ with no endpoint in $S$. Thus, $S$ is a vertex cover. Also, for each of the edges in $M$, one of the endpoints must be in every vertex cover. Therefore, $|S_{opt}| \geq |M|$. Since $2 \cdot |M| = |S|$, $|S| \leq 2 \cdot |S_{opt}|$.

One can state the weighted vertex cover problem as a special case of the weighted set-covering problem. Let $U = E$, and for each vertex $v$, let $S_v$ be the set of edges incident to $v$ in $G$, and its weight is $\omega(v)$. Since each edge is incident to, at most, two vertices, $f = 2$. Thus, Hochbaum's algorithm, as she points out, yields a solution of weight $W$, for the weighted VC problem, such that

$$W \leq 2 \cdot W_{\text{opt}}. \tag{5}$$

Our purpose is to bypass the need to use a solution of a linear program. In the next section we present an algorithm which yields a solution of the set-covering problem which also satisfies (4), and therefore, in the case of VC, satisfies (5). However, our algorithm does not use a solution of a linear program; in fact, its running time is linear in $\sum_{i=1}^{n}|S_i|$. Thus, in the case of VC its running time is linear in $|E|$.

## The Algorithm

Let us use the following notations.

The input is a family of sets $S_1, S_2, \ldots, S_n$, and $N = \{1,2,\ldots,n\}$. The universal set is $e_1, e_2, \ldots, e_t$ and $T = \{1,2,\ldots,t\}$. For each $j \in T$, $F(j) = \{i \mid e_j \in S_i\}$. The given weight of $S_i$ is $\omega_i$, while $SW(i)$ denotes the "residual" weight of $S_i$. For $j \in T$, $EW(j)$ is the "weight" assigned to $e_j$. $I$ denotes the set of indices of sets already selected to be in the cover, and will present the output when the algorithm halts; $J$ denotes the set of indices of elements not yet covered, and will be empty when the algorithm halts.

The statements in square brackets are not necessary for running the algorithm, but are useful in the proof that (4) holds.

```
 (1) $\forall i\ SW(i) \leftarrow \omega_i$, [$\forall j\ EW(j) \leftarrow 0$].
 (2) $I \leftarrow \emptyset$, $J \leftarrow T$.
 (3) Do while $J \neq \emptyset$
 (4) Let $j \in J$.
 (5) $M \leftarrow \mathrm{Min}(SW(i) \mid i \in F(j))$, [$EW(j) \leftarrow M$].
 (6) Let $k \in F(j)$ such that $SW(k) = M$.
 (7) $\forall i \in F(j)\quad SW(i) \leftarrow SW(i) - M$.
 (8) $I \leftarrow I \cup \{k\}$, $J \leftarrow J - S_k$.
 (9) end
(10) Stop.
```

THEOREM 1. *The time-complexity of the algorithm is linear in* $\sum_{i=1}^{n}|S_i|$.

*Proof.* Each time the algorithm goes through loop (4)–(8) it spends time proportional to $|F(j)|$, for steps (5)–(7), and time proportional to $|S_k|$ in step (8). However, each $j$ and each $k$ come up at most once, and $\sum_{j=1}^{t}|F(j)| = \sum_{i=1}^{n}|S_i|$. Q.E.D.

Clearly, upon termination, $I$ constitutes a set-cover, however the proof that (4) holds is more complicated.

THEOREM 2. *The total weight $W$ of the set-cover $I$, produced by the algorithm, satisfies*

$$W \le W_{\text{opt}} \cdot \operatorname*{Max}_{j \in T} |F(j) \cap I|. \tag{6}$$

This a posteriori bound clearly implies (4), and is the same as Hochbaum's.

*Proof.* Before we prove (6) let us prove several properties.

$$\forall i \in N, \qquad SW(i) \ge 0. \tag{7}$$

This is true throughout the algorithm, since in step (1), $SW(i) \leftarrow \omega_i$ and in step (7) we subtract from each $SW(i)$, $i \in F(j)$, the least of them.

$$\forall j \in T, \qquad EW(j) \ge 0. \tag{8}$$

In step (1), we start with $EW(j) \leftarrow 0$, and in step (5) we replace it by $M$, which by (7) is nonnegative.

$$\forall i \in N, \qquad SW(i) + \sum_{j \in S_i} EW(j) = \omega_i. \tag{9}$$

Initially, $SW(i) = \omega_i$ while $EW(j) = 0$. When $EW(j)$ changes to $M$ (in step (5)), and if $j \in S_i$ then in step (7), $SW(i)$ is reduced by the same amount and (9) is restored.

$$\forall i \in N, \qquad \sum_{j \in S_i} EW(j) \le \omega_i. \tag{10}$$

This follows immediately, from (9) and (7).

$$\forall i \in I, \qquad \sum_{j \in S_i} EW(j) = \omega_i. \tag{11}$$

This follows from (9) and the fact that when $i$ enters $I$ (denoted by $k$ in the algorithm) then by steps (6) and (7), $SW(i)$ becomes zero.

$$W \le \sum_{j \in T} EW(j) \cdot \operatorname*{Max}_{j \in T} |F(j) \cap I|. \tag{12}$$

This can be proved as follows.

$$W = \sum_{i \in I} \omega_i \qquad \text{(by definition)}$$

$$= \sum_{i \in I} \sum_{j \in S_i} EW(j) \qquad \text{(by (11))}.$$

In this summation $EW(j)$ is counted $|F(j) \cap I|$ times. Thus (12) holds. Now,

$$\sum_{j \in T} EW(j) \leq \text{Max} \sum_{j \in T} y_j, \tag{13}$$

where the $y_j$'s satisfy the following constraints:

$$\forall i \in N, \qquad \sum_{j \in S_i} y_j \leq \omega_i,$$

$$\forall j \in T, \qquad y_j \geq 0.$$

This follows from the fact that the $EW(j)$'s satisfy the constraints, by (10) and (8).

Consider now the (dual) program: Minimize $\Sigma_{i \in N} x_i \cdot \omega_i$ subject to the constraints

$$\forall j \in T, \qquad \sum_{i \in F(j)} x_i \geq 1,$$

$$\forall i \in N, \qquad x_i \geq 0.$$

Let us show that

$$\text{Max} \sum_{j \in T} y_j \leq \text{Min} \sum_{i \in N} x_i \cdot \omega_i. \tag{14}$$

This (duality property) follows from

$$\sum_{j \in T} y_j \leq \sum_{j \in T} y_j \sum_{i \in F(j)} x_i = \sum_{i \in N} x_i \sum_{j \in S_i} y_j \leq \sum_{i \in N} x_i \cdot \omega_i.$$

Let $I_{\text{opt}}$ specify an optimal set-cover, i.e.,

$$W_{\text{opt}} = \sum_{i \in I_{\text{opt}}} \omega_i. \tag{15}$$

If we define

$$x_i = 1 \qquad \text{if } i \in I_{\text{opt}},$$
$$x_i = 0 \qquad \text{if } i \notin I_{\text{opt}},$$

then clearly these $x_i$'s satisfy the constraints (by $I_{\text{opt}}$ being a set-cover) and therefore

$$W_{\text{opt}} \geq \text{Min} \sum_{i \in N} x_i \cdot \omega_i. \tag{16}$$

By (13), (14), and (16)

$$\sum_{j \in T} EW(j) \leq W_{\text{opt}},$$

and by (12) the theorem follows. Q.E.D.

Clearly, our proof uses techniques of linear programming (i.e., duality theory), but our algorithm does not involve a solution of a linear program.

The bound given by (4), for the performance of our algorithm, is tight. The following simple example establishes this fact.

$$\begin{aligned} S_1 &= \{e_1\}, \\ S_2 &= \{e_1, e_2\}, \\ S_3 &= \{e_1, e_3\}, \\ &\vdots \\ S_{n-1} &= \{e_1, e_{n-1}\}, \\ S_n &= \{e_1, e_2, e_3, \ldots, e_n\}, \\ \omega_1 &= \omega_2 = \cdots = \omega_{n-1} = \omega_n = a. \end{aligned}$$

If the algorithm picks $j = 1$ as its first value, and $j = n$ as its last, all sets may be in the resulting cover. Thus, $W = n \cdot a$, while $W_{\text{opt}} = a$ and $f = n$.

## ACKNOWLEDGMENT

The authors would like to thank Azaria Paz and Tom Hoholdt for their helpful comments.

## REFERENCES

1. R. M. KARP, Reducibility among combinatorial problems, *in* "Complexity of Computer Computations," (R. E. Miller and J. W. Thatcher, Eds.), pp. 85–104, Plenum, New York, 1972.
2. M. R. GAREY AND D. S. JOHNSON, "Computers and Intractability: A Guide to the Theory of NP-Completeness," Freeman, San Francisco, 1979.
3. S. EVEN, "Graph Algorithms," Computer Science Press, Washington, D.C., 1979.
4. V. CHVÁTAL, A greedy-heuristic for the set-covering problem, *Math. Operations Res.* 4, No. 3 (1979), 233–235.
5. D. S. HOCHBAUM, Approximation algorithm for the weighted set covering and node cover problems, unpublished manuscript, April 1980.
6. B. ASPVALL AND R. E. STONE, Khachiyan's linear programming algorithm, *J. Algorithms* 1, No. 1 (1980), 1–13.