MATHEMATICS OF OPERATIONS RESEARCH
Vol. 4, No. 3, August 1979
Printed in U.S.A.

A GREEDY HEURISTIC FOR THE SET-COVERING PROBLEM*

V. CHVATAL

McGill University

Let A be a binary matrix of size $m \times n$, let c^T be a positive row vector of length n and let e be the column vector, all of whose m components are ones. The set-covering problem is to minimize c^Tx subject to $Ax \geqslant e$ and x binary. We compare the value of the objective function at a feasible solution found by a simple greedy heuristic to the true optimum. It turns out that the ratio between the two grows at most logarithmically in the largest column sum of A. When all the components of c^T are the same, our result reduces to a theorem established previously by Johnson and Lovász.

In the *set-covering problem* [2], the data consist of finite sets $P_1, P_2, \ldots, P_n$ and positive numbers $c_1, c_2, \ldots, c_n$. We denote $\bigcup(P_j : 1 \leqslant j \leqslant n)$ by I and write $I = \{1, 2, \ldots, m\}$, $J = \{1, 2, \ldots, n\}$. A subset J^* of J is called a *cover* if $\bigcup(P_j : j \in J^*) = I$; the *cost* of this cover is $\sum(c_j : j \in J^*)$. The problem is to find a cover of minimum cost.

The set-covering problem is notoriously hard; in fact, it is known to be *NP*-complete [4], [1]. In view of this fact, the relative importance of heuristics for solving the set-covering problem increases. The purpose of this note is to establish a tight bound on the worst-case behaviour of a rather straightforward heuristic. In case $c_j = 1$ for all j, our theorem reduces to one obtained previously by Johnson [3] and Lovász [5].

Intuitively, it seems that the desirability of including j in an optimal cover increases with the ratio $|P_j|/c_j$ which counts the number of points covered by P_j per unit cost. This sentiment suggests a recursive procedure for finding near-optimal covers.

Step 0. Set $J^* = \emptyset$.

Step 1. If $P_j = \emptyset$ for all j then stop: J^* is a cover. Otherwise find a subscript k maximizing the ratio $|P_j|/c_j$ and proceed to Step 2.

Step 2. Add k to J^*, replace each P_j by $P_j - P_k$ and return to Step 1.

Heuristic procedures of a similar character are called *greedy*.

For illustration, consider sets $P_1, P_2, \ldots, P_{m+1}$ and numbers $c_1, c_2, \ldots, c_{m+1}$ such that $P_j = \{j\}$ and $c_j = 1/j$ for $j = 1, 2, \ldots, m$ whereas $P_{m+1} = I$ and $c_{m+1} > 1$. Our greedy heuristic returns $J^* = \{1, 2, \ldots, m\}$, the winning ratio in iteration r being $|P_{m+1-r}|/c_{m+1-r} = m + 1 - r$. The cost of J^* is

$$H(m) = \sum_{j=1}^{m} \frac{1}{j}.$$

However, $\{m + 1\}$ is also a cover and its cost c_{m+1} can be arbitrarily close to 1. Thus the cost of the cover returned by the greedy heuristic can exceed the cost of an optimal cover by a factor arbitrarily close to $H(m)$. On the other hand, we shall show that the factor never exceeds $H(m)$. In fact, the upper bound can be improved into $H(d)$ such that d is the size of the largest set P_j.

* Received January 30, 1978.
AMS 1970 subject classification. Primary 90C10. Secondary 05B40, 52A45.
IAOR 1973 subject classification. Main: Nonlinear programming. Cross reference: Sets.
Key words. Set covering, integer programming, evaluation of greedy heuristic.

0364-765X/79/0403/0233$01.25

THEOREM. *The cost of the cover returned by the greedy heuristic is at most $H(d)$ times the cost of an optimal cover.*

We shall prove a stronger but less concise result. Define an $m \times n$ matrix $A = (a_{ij})$ by

$$a_{ij} = \begin{cases} 1 & \text{if } i \in P_j, \\ 0 & \text{otherwise,} \end{cases}$$

so that the n columns of A are the incidence vectors of $P_1, P_2, \ldots, P_n$. Clearly, the incidence vector $x = (x_j)$ of an arbitrary cover satisfies

$$\sum_{j=1}^{n} a_{ij} x_j \geqslant 1 \quad \text{for all } i,$$

$$x_j \geqslant 0 \quad \text{for all } j.$$

We claim that these inequalities imply

$$\sum_{j=1}^{n} H\left(\sum_{i=1}^{m} a_{ij} \right) c_j x_j \geqslant \sum (c_j : j \in J^*) \tag{1}$$

for the cover J^* returned by the greedy heuristic. Once (1) is proved, the theorem will follow by letting x be the incidence vector of an optimal cover.

To prove (1), it will suffice to exhibit nonnegative numbers $y_1, y_2, \ldots, y_m$ such that

$$\sum_{i=1}^{m} a_{ij} y_i \leqslant H\left(\sum_{i=1}^{m} a_{ij} \right) c_j \quad \text{for all } j \tag{2}$$

and such that

$$\sum_{i=1}^{m} y_i = \sum (c_j : j \in J^*), \tag{3}$$

for then

$$\sum_{j=1}^{n} H\left(\sum_{i=1}^{m} a_{ij} \right) c_j x_j \geqslant \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_i \right) x_j = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \right) y_i$$

$$\geqslant \sum_{i=1}^{m} y_i = \sum (c_j : j \in J^*)$$

as desired.

The numbers $y_1, y_2, \ldots, y_m$ satisfying (2) and (3) have a simple intuitive interpretation: each y_i is the price paid by the greedy heuristic for covering the point i. To make this definition more precise, let us denote by P_j^r the set P_j at the beginning of iteration r; for typographical simplicity, we shall denote the size of P_j^r by w_j^r. Without loss of generality, we may assume that J^* is $\{1, 2, \ldots, r\}$ after r iterations, and so

$$w_r^r / c_r \geqslant w_j^r / c_j$$

for all r and j. If there are t iterations altogether then

$$\sum (c_j : j \in J^*) = \sum_{j=1}^{t} c_j.$$

Observe that each $i \in I$ belongs to precisely one of the sets P_r^r with $r = 1, 2, \ldots, t$.

For this r, we have

$$y_i = c_r / w_r^r.$$

Now (3) becomes a triviality: we have

$$\sum_{i=1}^{m} y_i = \sum_{r=1}^{t} \Sigma(y_i : i \in P_r^r) = \sum_{r=1}^{t} w_r^r (c_r / w_r^r) = \sum_{r=1}^{t} c_r.$$

To prove (2), observe that $P_j \cap P_r^r = P_j^r - P_j^{r+1}$ and so

$$\sum_{i=1}^{m} a_{ij} y_i = \sum_{r=1}^{t} \Sigma(y_i : i \in P_j \cap P_r^r)$$

$$= \sum_{r=1}^{t} \left(w_j^r - w_j^{r+1}\right) \cdot (c_r / w_r^r).$$

If s is the largest superscript such that $w_j^s > 0$ then

$$\sum_{i=1}^{m} a_{ij} y_i = \sum_{r=1}^{s} \left(w_j^r - w_j^{r+1}\right) \cdot (c_r / w_r^r)$$

$$\leqslant c_j \sum_{r=1}^{s} \left(w_j^r - w_j^{r+1}\right) / w_j^r.$$

The rest is a routine manipulation: we have

$$\sum_{r=1}^{s} \left(w_j^r - w_j^{r+1}\right) / w_j^r \leqslant \sum_{r=1}^{s} \left(H(w_j^r) - H\left(w_j^{r+1}\right)\right) = H\left(w_j^1\right)$$

and, of course,

$$w_j^1 = |P_j| = \sum_{i=1}^{m} a_{ij}.$$

The author is indebted to Roy Marsten and to an anonymous referee for helpful suggestions which led to an improved presentation of this note.

References

[1] Aho, A. V., Hopcroft, J. E. and Ullman, J. D. (1974). *The Design and Analysis of Computer Algorithms.* Addison-Wesley, Reading, Mass.

[2] Garfinkel, R. S. and Nemhauser, G. L. (1972). *Integer Programming.* John Wiley & Sons, New York.

[3] Johnson, D. S. (1974). Approximation Algorithms for Combinatorial Problems. *J. Comput. System Sci.* **9** 256–278.

[4] Karp, R. M. (1972). Reducibility among Combinatorial Problems. In: *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, eds. Plenum Press, New York.

[5] Lovász, L. (1975). On the Ratio of Optimal Integral and Fractional Covers. *Discrete Math.* **13** 383–390.

SCHOOL OF COMPUTER SCIENCE, MCGILL UNIVERSITY, MONTREAL, CANADA .