CENTER FOR
PARALLEL OPTIMIZATION

BALANCED SOLUTION TO A CLASS OF GENERALIZED SET COVERING

by

R. Cerulli, R. De Leone, M. Gaudioso and R. Mautone

Computer Sciences Technical Report #969

September 1990

BALANCED SOLUTION TO A CLASS OF GENERALIZED SET COVERING

R. CERULLI*, R. DE LEONE†, M. GAUDIOSO‡ AND R. MAUTONE*

Abstract. In this paper we consider a generalization of the set covering problem that has applications in manpower scheduling and in multiple cover system design for air traffic control and defense. The heuristic technique we propose calculates well-balanced approximate solutions to the problem. The worst–case behavior of the algorithm for a special case is analyzed.

Key words. heuristic, set covering

1. Introduction. In this paper we are interested in analyzing an approximation scheme to the solution of the system of linear inequalities:

$$\begin{aligned} Ax &\geq b \\ e^T x &= n \\ x &\geq 0 \text{ and integer} \end{aligned} \tag{1.1}$$

where A is a binary matrix of dimension mxp, b is a m-dimensional vector with positive integer components, e is a vector of ones and n is a given positive integer. The problem we investigate is a generalization of the problem of finding a feasible solution for the set covering problem. A vast literature has been devoted to the solution of set covering problems especially in view of the large number of real–world applications of these problems (see [6, 4] for a survey of the results in this area). In [1] various techniques based on combination of different approaches are presented for the solution of the classical set covering problem. In [2, 3] the behavior of greedy algorithms for this class of problems is discussed. Our formulation differs from the classical formulation of set covering feasibility problem in two important features:

1. The right-hand-side vector b is a positive integer vector (not necessarily with unit components). This corresponds to a "multiple" cover for a given set that is extremely important in many practical applications (for example in manpower schedule creation, high reliability systems, defense systems, etc.)
2. The possibility of replicating a particular column of A in the covering obtained by using integer variables instead of binary variables.

In a previous paper [5] a similar problem has been analyzed in the context of optimal manpower scheduling.

In this paper we will present a new greedy algorithm for determining a "balanced" solution of (1.1). The key idea of the heuristic is that, at each step, a new column of the matrix A is chosen in such a way that the largest value of the vector $s - b$ (where s is the sum of the already selected columns of the matrix A) is close to the smallest value of the same vector.

* Dipartimento di Informatica e Applicazioni, Università di Salerno, Salerno, Italy.

† Center for Parallel Optimization, Computer Sciences Department, University of Wisconsin Madison, 1210 West Dayton Street, Madison, WI 53706. The work of this author has been supported by the Air Force Office of Scientific Research Grant AFOSR-89-0410.

‡ Dipartimento di Sistemi, Università della Calabria, Rende, Cosenza, Italy.

In the next section alternative variants of problem (1.1) will be discussed and the heuristic will be defined. In Section 3 an analysis of the worst–case behavior for the algorithm is presented. Finally in Section 4 we will discuss some alternative formulation of the problem and modifications to the proposed heuristic.

In our notation, a superscript T will denote the transpose and the inner product of two vector z and w in $\mathbb{R}^k$ will be denoted by $z^T w$. For a mxp matrix A, A_j will denote the jth column of A and A_{ij} the element of A in row i and column j. The ith component of a vector x in $\mathbb{R}^n$ will be denoted by x_i. The symbol := denotes definition of the term on the left of the symbol.

2. Definition of the heuristic. In many practical cases, the objective represents the best use of the available resources (in our case, the choice of the n columns of the matrix A, with some of the columns chosen more than once) with the goal of obtaining a cover as close as possible to the target indicated by the vector b. In these cases, it seems useful to formulate an optimization problem in the form

$$\begin{array}{lc} \text{minimize} & \nu_{\max} - \nu_{\min} \\ \text{subject to} & \nu_{\min} e \leq Ax - b \leq \nu_{\max} e \\ & e^T x = n \\ & x \geq 0 \text{ and integer} \end{array} \tag{2.1}$$

The above problem is equivalent to the following problem:

$$\begin{array}{lc} \text{minimize} & \chi(x) := \left(\max\limits_{i=1,\dots,m} (Ax - b)_i \right) - \left(\min\limits_{i=1,\dots,m} (Ax - b)_i \right) \\ \text{subject to} & e^T x = n \\ & x \geq 0 \text{ and integer} \end{array} \tag{2.2}$$

The function $\chi(x)$ is a piecewise linear convex function since it is the difference of a convex function and a concave function. In the above formulation (2.1) the variables $\nu_{\max}$ and $\nu_{\min}$ are not restricted in sign. The solution obtained is, therefore, the most balanced solution but does not necessarily satisfy $Ax - b \geq 0$ (even if a feasible solution of this system of inequalities exists). Different formulations can be obtained by requiring $\nu_{\min} \geq 0$ or $\nu_{\max} \geq 0$ and will be discussed in Section 4.

It should be noticed that the solution provided by the above model (2.1) is different from the solution obtained by choosing as objective function the Chebyshev norm of $Ax-b$, $\|Ax - b\|_\infty := \max_{i=1,\dots,n} |(Ax - b)_i|$. In this case the largest component (in modulus) of the vector $Ax - b$ will be minimized, without any attempt to obtain a well balanced solution.

The heuristic we propose for problem (2.1) is similar to the heuristic described, in a different context, in [5]. First, note that the problem requires to select, in an optimal way, n columns, not necessarily distinct, of A. The proposed greedy algorithm will choose a column at the time from the matrix A based on a local merit function with the goal of making the vector $s - b$ as balanced as possible, where s is the sum of the columns of A already selected. At the first step, the vector s is set equal to zero. At the generic step, we define a merit function $\phi(j),\ j = 1, \ldots, p$ and associate a value to each column j in the following way. First, the minimum and the maximum values of $s - b$ are computed

$$\nu_{\min} := \min_{i=1,\ldots,m} \{s_i - b_i\} \tag{2.3}$$

$$\nu_{\max} := \max_{i=1,\ldots,m} \{s_i - b_i\} \tag{2.4}$$

and the set L and U are defined:

$$L := \{i : s_i - b_i = \nu_{\min}\}$$

$$U := \{i : s_i - b_i = \nu_{\max}\}$$

Finally the function $\phi(j)$ is defined as

$$\phi(j) := \underline{\phi}(j) + \overline{\phi}(j) \tag{2.5}$$

where

$$\underline{\phi}(j) := \sum_{i \in L} A_{ij} \qquad \overline{\phi}(j) := \sum_{i \in U} (1 - A_{ij})$$

This merit function corresponds to assigning, for each column, unitary value to the ones in row corresponding to minimal components of $s - b$ and to the zeroes corresponding to maximal components of $s - b$. The obvious choice is to select at every iteration a column j^* such that

$$j^* = \operatorname*{argmax}_{j=1,\ldots,p} \phi(j).$$

We are now ready to describe completely our algorithm:

Greedy algorithm

begin
    set $s = 0$, $k = 0$, $x = 0$
    while $k < n$ **do**
        begin
            compute $\phi(j), j = 1, \ldots, p$
            find $j^* = \operatorname*{argmax}_{j=1,\ldots,p} \phi(j)$
            set $s = s + A_{j^*}$, $x_{j^*} = x_{j^*} + 1$, $k = k + 1$
        end
end

It is easy to show that the complexity of the algorithm is $O(nmp)$.

Remark. The proposed merit function assigns equal weights to both functions $\underline{\phi}(j)$ and $\overline{\phi}(j)$ and therefore tries to construct the most balanced solution (not necessarily a feasible cover). This is in the spirit of problem (2.1) where the difference between the largest and the smallest component of $Ax - b$ is minimized. It should be also noticed that the maximum value of the merit function $\phi(j)$ can be achieved by more than one value of j. Therefore some tie-breaking mechanism must be implemented in choosing the column index j^*.

□

3. Worst–case Analysis. In this section we will analyze the worst–case behavior of the greedy algorithm. More specifically, we will discuss the behavior of the algorithm for the following problem:

> determine the minimum value of n such that $e^T x = n, x \geq 0$, integer and there exists a feasible solution of the system of inequalities $Ax \geq b$.

The following assumptions will be made on the problem:

1. the vector b has all positive identical component (that is $b = re$ for some positive integer r);
2. all columns of A have the same number τ of nonzero entries;
3. at every iteration $\phi(j^*) = \max_{j=1,\dots,p} \phi(j) \geq 2$ and $\underline{\phi}(j^*) \geq 1, \overline{\phi}(j^*) \geq 1$

Note that the last condition is verified if the columns of A are all the cyclic permutation of the same nonperiodic vector. In [5] a class of problems where these assumptions are satisfied is presented. In [5] the vector b represents the workload request for a particular shift and the assumption $b = re$ corresponds to the case of uniform staffing that is frequent in many applications.

The following proposition provides an upper bound for the value n computed by the greedy algorithm. In view of Assumption 1, will be sufficient to consider the behavior of the vector s instead of $s - b$.

PROPOSITION 3.1. *Suppose that the Assumptions 1, 2 and 3 are satisfied. Then the greedy algorithm provides a solution x^* of the system of inequalities $Ax \geq b$ with $e^T x^* \leq r(m - \tau) + 1$*

Proof. After $m - \tau + 1$ iterations, taking into account that $\underline{\phi}(j^*) \geq 1$, we have that

$$s_{min} := \min_{i=1,\dots,m} \{s_i\} \geq 1.$$

Obviously we also have that

$$s_{max} := \max_{i=1,\dots,m} \{s_i\} \leq m - \tau + 1.$$

It should be noticed that $|L| \leq m - \tau$ (otherwise, all remaining elements of s should be greater than or equal to $m - \tau + 1$ that is impossible since $\overline{\phi}(j^*) \geq 1$. The two above observations imply that, in the worst case, after $m - \tau + 1$ iterations at most $m - \tau$ components of s have a minimum value equal to 1, all remaining components having larger values. Therefore, we can conclude that after $r(m - \tau) + 1$ iterations, the minimum value of s is at least r. □

In order to evaluate the maximum error, we observe that, (see [5]) under the Assumptions 1, 2 and 3, $\lceil rm/\tau \rceil$ provides a lower bound on n. Therefore, the maximum relative error is:

$$e\text{max} = \frac{r(m-\tau) + 1 - rm/\tau}{rm/\tau}.$$

For sufficiently large values of rm/τ:

$$e_{max} \simeq (\tau - 1) - \frac{\tau^2}{m}.$$

Note that, for any choice of m, the maximum error is obtained for $\tau = \frac{m}{2}$

4. Alternative problem formulation. In the section we will discuss some possible alternative formulations to problem (2.1) and modifications to the greedy algorithm presented in Section 2. As already noted, the solution obtained from (2.1) is the most balanced solution (i.e. the difference between the largest and the smallest component in $Ax - b$ is minimum). However the optimal solution is not necessarily feasible and in many cases it is extremely important to obtain a feasible or "almost feasible" cover. We will show that the feasibility (or "almost feasibility") condition can be enforced by adding the constraint $\nu_{\min} \geq 0$ or the constraint $\nu_{\max} \geq 0$. In the first case (i.e if we add the constraint $\nu_{\min} \geq 0$ to the problem formulation (2.1)), we restrict our attention to only feasible solutions and among these solutions we select the most balanced. The drawback of this approach is that the feasible region can be empty.

In the second case the problem has the form:

$$(4.1) \qquad \begin{array}{ll} \text{minimize} & \nu_{\max} - \nu_{\min} \\ \text{subject to} & \nu_{\min} e \leq Ax - b \leq \nu_{\max} e \\ & e^T x = n \quad \nu_{\max} \geq 0 \\ & x \geq 0 \text{ and integer} \end{array}$$

In this formulation, negative values of $\nu_{\min}$ are penalized by the objective function and since the difference between $\nu_{\max}$ (that caanot assume negative values) and $\nu_{\min}$ is minimized, the error in satisfying the covering constraint is also minimized. Note that in this case the problem is equivalent to the minimization of a piecewise linear convex function:

$$(4.2) \qquad \begin{array}{ll} \text{minimize} & \chi'(x) := \left(\max\{0, \max_{i=1,\dots,m} (Ax - b)_i\}\right) - \left(\min_{i=1,\dots,m} (Ax - b)_i\right) \\ \text{subject to} & e^T x = n \\ & x \geq 0 \text{ and integer} \end{array}$$

We now indicate how to construct an alternative merit function that will take into account the new constraint imposed on $\nu_{\max}$. First, redefine the quantity (2.4) and let

$$\nu_{\max} := max\{0, \max_{i=1,\dots,m}\{s_i - b_i\}\}$$

Define $\nu_{\min}$, the set L and U and the quantities $\underline{\phi}(j)$ and $\overline{\phi}(j)$ as in Section 2. Finally, set

$$\phi^{\lambda}(j) := \lambda\underline{\phi}(j) + (1-\lambda)\overline{\phi}(j)$$

where λ is a parameter varying between 0 and 1. Choosing $\lambda = 1$ the greedy algorithm of Section 2 (with the new merit function) will attempt to construct a feasible cover (but not necessarily a balanced cover). In fact for $\lambda = 1$ we favor columns of A with ones corresponding to minimal components of $s - b$. If, instead, we chose $\lambda = 0$ columns of A that, if chosen, will produce an overshooting of the cover are penalized. Finally, if $\lambda = 0.5$ the greedy algorithm will produce an approximate solution to (4.1).

5. Conclusions. A simple heuristic technique has been discussed for the generalized set covering feasibility problem. The algorithm produces a balanced solution with the goal of obtaining a cover as close as possible to the target value. A worst–case analysis for a special structured case has been also presented.

REFERENCES

[1] E. BALAS AND A. HO, *Set covering algorithm using cutting planes, heuristics and subgradient optimization: a computational study*, Mathematical Programming Study, 12 (1980), pp. 37–60.

[2] V. CHVATAL, *A greedy heuristic for the set covering problem*, Mathematics of Operations Research, 4,3 (1979), pp. 233–235.

[3] D. S. JOHNSON, *Approximation algorithms for combinatorial problems*, Journal of Computer and System Science, 9 (1974), pp. 256–278.

[4] G.L. NEMHAUSER AND L. A. WOLSEY, *Integer and Combinatorial Optimization*, John Wiley & Sons, 1988.

[5] M. GAUDIOSO R. CERULLI AND R. MAUTONE, *A class of manpower scheduling problems*, Zeitschrift für Opns. Res., to appear.

[6] H.M. SALKIN AND K. MATHUR, *Foundations of integer programming*, North–Holland, 1989.