Revue Belge de Statistique, d'Informatique et de Recherche Opérationnelle, Vol 22, *n°* 2
Belgisch Tijdschrift voor Statistiek, Informatica en Operationeel Onderzoek, Vol. 22, *Nr* 2.

A CLASS OF LOCATION, DISTRIBUTION AND SCHEDULING PROBLEMS : MODELING AND SOLUTION METHODS

TUTORIAL PAPER X

by

Egon Balas
Carnegie-Mellon University
Pittsburgh, Pa., U.S.A.

Paper presented at the Chinese-American Symposium on Systems Analysis and Engineering, Xian, April 1981

The research underlying this paper was supported by Grant ECS- 7902506 of the National Science Foundation, and Contract N00014-75-C-0621 NR 047-048 with the U.S. Office of Naval Research.

1. Introduction

A large variety of location, distribution, scheduling and other prob- ns can be formulated as variants of a mathematical model known as the ering or set covering problem. A partial list of real world problems to ch this approach has been successfully applied includes the following stances (see the Bibliography on Applications at the end of the paper) :

- site selection and facility location-allocation problems
- location of emergency service facilities (fire stations, hospitals etc.)
- choice of size and location of drilling platforms in offshore oilfields
- vehicle routing : truck dispatching problem, tanker fleet and airline fleet scheduling
- crew scheduling for airlines, bus companies, railways
- the minimum test set (diagnostic) problem (in industry, medecine, experimental design)
- switching circuit design (electrical engineering)
- distribution of broadcasting frequencies among radio or TV stations
- information retrieval (from computer files)
- assembly line balancing
- stock cutting
- various capital investment decisions

The (weighted) set covering problem is

$$\text{(SC)} \qquad \min\{cx \mid Ax \geq e,\ x_j = 0 \text{ or } 1,\ j \in N\}$$

where $A = (a_{ij})$, $a_{ij} = 0$ or 1, $i \in M = \{1,\ldots,m\}$, $j \in N = \{1,\ldots,n\}$, $c \in R^n$, $e \in R^m$, and $e = (1,\ldots,1)$. Its name comes from the following interpretation: if the rows of A are associated with the elements of the set M, and each column a_j of A with the subset M_j of those $i \in M$ such that $a_{ij} = 1$, then (SC) is the problem of finding a minimum-weight family of subsets M_j, $j \in N$, whose union is M, i.e., which "cover" M, each subset M_j being weighted with c_j. The special case when $c_j = 1$, $\forall\, j \in N$, is called the simple (unweighted) covering problem.

Another interpretation of (SC) is as follows. Let $G = (V, E)$ be a bipartite graph, i.e., a graph whose vertex set V can be partitioned into two subsets, V_1 and V_2, such that $E \subseteq V_1 \times V_2$, i.e., every edge $(i, j) \in E$ is of the form $i \in V_1$, $j \in V_2$. We say that a vertex j in V_2 covers a vertex i in V_1 if $(i, j) \in E$. If vertex j has weight c_j, $\forall\, j \in V_2$, then (SC) is the problem of covering the vertices of V_1 with a minimum-weight subset of the vertices of V_2, with $M = V_1$, $N = V_2$, and for $j \in V_2$, $M_j = \{i \in V_1 \mid (i, j) \in E\}$.

A close relative of (SC) is (weighted) set partitioning (equality-constrained set covering) problem

$$\text{(SP)} \qquad \min\{cx \mid Ax = e,\ x_j = 0 \text{ or } 1,\ j \in N\}$$

where A, e and c are as before. (SP) can be brought to the form (SC) by writing

$$\min\{cx + \theta ey \mid Ax - y = e,\ y \geq 0,\ x_j = 0 \text{ or } 1,\ j \in N\}$$

and then, using $y = Ax - e$,

$$\min\{-\theta m + c'x \mid Ax \geq e,\ x_j = 0 \text{ or } 1,\ j \in N\}$$

with $c' = c + \theta eA$. For sufficiently large θ (for instance, $\theta > \sum_{j \in N} c_j$), this problem has the same set of optimal solutions as (SP) whenever the latter is feasible. Both set covering and set partitioning are used in formulating the problems listed above, and sometimes a mixed covering-partitioning problem arises. Also, in many real-world situations a few extra constraints may be needed, which require appropriate modifications of the solution methods.

In the next section we discuss the modeling potential of the set covering approach, illustrating the problem formulation techniques on several important classes of real-world problems. In section 3 we describe a class of algorithms for solving set covering problems, based on cutting planes, heuristics and subgradient optimization. Finally, as an Appendix we provide two bibliographies, one on theory and algorithms, the second one on applications (classified by area) of the set covering and set partitioning models.

2. Modeling Techniques

The high versatility of the model under discussion stems from the fact that all the real world problems listed above, and a great variety of other problems, can be formulated as follows. Given

(i) a finite set M;

(ii) a system of constraints on the elements of M, defining a family F of "acceptable" subsets of M; and

(iii) a function on M defining a cost for every member of the family F;

find a minimum-cost collection of members of F which cover M, i.e., whose union is M.

The applicability of the set covering model to problems amenable to this formulation is based on the simple but important observation that in most cases problems of this form can be solved with a satisfactory degree of precision by the following two-stage approximation procedure.

Stage 1. Using (ii) and (iii), generate explicitly a subfamily $\hat{F} \subseteq F$, $\hat{F} = \{M_j\}_{j \in N}$, with associated costs c_j, $j \in N$, for which the probability that $\hat{F}$ contains an optimal solution is sufficiently large.

Stage 2. Replace the objective function (iii) by cx, and the system of constraints (ii) by $Ax \geq e$, $x_j = 0$ or 1, $j \in N$, where the columns of A correspond to the elements of $\hat{F}$ (i.e., $a_{ij} = 1$ if $i \in M_j$ and $a_{ij} = 0$ otherwise), and solve the resulting set covering problem.

In the following we illustrate this modeling procedure on several examples.

Offshore Drilling Platforms. To start the exploration of an offshore oilfield, after fixing the location of the wells to be drilled on the basis of geological data, one has to choose the appropriate size and location of the platforms to be used for the drilling (and later for the exploitation) of the wells. Drilling platforms vary immensely in size and cost. A platform may handle just one or two wells, or as many as 30-40 wells; it can be just a few yards high or as high as the Empire State Building; and it may cost anywhere between a few hundred thousand dollars and 100 million dollars. The best platform/well configuration depends on the distances between the wells, the shape of the seabed, the depth of the water, the depth to which one has to drill to reach the oil, etc. These factors define both the system of constraints on the size and location of the platforms, and the cost function. Rather than trying to write down explicitly these complicated and highly nonlinear functions and constraints, one can proceed as follows. Given the location of m wells expressed as a set of coordinate pairs in 2-space,

$(w_{11}, w_{12}),\ldots,(w_{m1}, w_{m2})$, a set of heuristic rules are defined and put into a computer program, for grouping together wells that might lend themselves to being drilled from a single platform. For each such group of wells, say M_j, the cost of the corresponding platform, connecting pipes and other necessary equipment, is estimated and expressed as a single number c_j. The wells are grouped in many different ways, and each group M_j corresponds to a candidate platform, i.e., one that may or may not be built. Each candidate platform j is associated with a cost c_j and a column a_j of a 0-1 matrix to be used in a set covering problem, namely $a_{ij} = 1$ if well i is included in the group of candidate platform i, $a_{ij} = 0$ otherwise. Solving the set covering (or set partitioning) problem formulated this way will then select an optimal combination of drilling platforms to be built. Although the set covering problem can usually (i.e., up to 1000-2000 columns) be solved to optimality, the solution obtained is not necessarily optimal for the real problem, since some combination of platforms and wells may have been omitted in the Stage 1 procedure of generating platform candidates. But if a sufficiently reliable procedure is used in Stage 1, i.e., one that does not omit any promising candidate, then the optimal solution of the Stage 2 problem should be pretty close to the optimum of the real problem.

Location and Number of Emergency Service Facilities. In deciding upon the number and location of hospitals, fire stations or other emergency service facilities dedicated to fill the needs of a certain area, one good criterion to use is that each point in the area be reachable from at least one facility in no more than some predetermined time limit t. If the points of the area (population centers, villages, quarters of a city, etc.) are represented as vertices of a graph, and the candidates for the location of a service facility as a subset of those vertices, and if the edges of the graph have lengths associated with them that reflect the time needed to reach an end-vertex of the edge from the other, then Stage 1 consists of determining, for each candidate location j, the set M_j of vertices reachable from j within the time limit t. Stage 2 will then solve a set covering problem whose coefficient matrix A has a column a_j for each candidate location for a service facility, with $a_{ij} = 1$ if point i can be reached from candidate facility j in no more time than t, $a_{ij} = 0$ otherwise. The solution gives both the number and location of the facilities needed. Obviously, the result is a function of the time limit t, and solving the problem for the relevant range of values of t also provides information about the cost of improving the emergency services, or the savings achievable through a relaxation of the service requirements.

Crew Scheduling. Airlines, bus companies, railways are facing the problem of scheduling their crews for the flights or trips to be provided in a given time period. To take the case of an airline, crews based in various cities have to be scheduled to man the flights of a given time period, say a week, so as to make the best use of their time. The conditions that have to be met are those of avoiding conflicts in the schedule, providing for reasonable breaks between flights, keeping a limit on the number of hours flown at night as well as a balance between the various crews in this respect, having each crew spend time periodically at its home base, etc. All these and other considerations that have to be taken into account give rise to a highly complex cost function and constraints, hard even to formulate. Instead of trying to do so, however, one usually sets up this problem as a set covering problem without ever writing down the constraints in functional form. In Stage 1, tentative routes are explicitly generated for each crew, that take into account the requirements, i.e., exclude conflicts, provide for breaks, etc. This is done by many airlines through heuristic programs that examine explicitly a very large number of possible schedules for each crew and retain those among them that are not obviously bad. Each such candidate schedule for a crew generates a column a_j of the 0-1 matrix A, where $a_{ij} = 1$ if schedule j (for a given crew) includes flight-leg i, $a_{ij} = 0$ otherwise, and a cost c_j which is a synthetic expression of the extent to which schedule j meets (or violates) the above listed requirements. Here a flight-leg is a flight leaving a given city at a given time and reaching another city at a given time. Solving the set covering problem (or set partitioning problem; depending on conditions specific to each airline), provides a schedule for each crew that covers all the flight legs to be covered during the period in question, while minimizing the total cost of the overall schedule (in terms of inconvenience, or sometimes actual money).

<u>The Minimum Test Set Problem</u>. The following problem arises in environments as diverse as product classification and quality control in industry, medical diagnostics, design of experiments, etc. Given a set of objects $Q = \{1,\ldots,q\}$, and a set of attributes (properties) $P = \{1,\ldots,p\}$ of some of these objects, find a minimal set S of properties to enable one to distinguish between the objects; in other words, find a set $S \subseteq P$ such that for every pair of objects $i, j \in Q$, there exists at least one property $k \in S$ such that object i has property k and object j does not have it, or vice versa. In an industrial context, the properties in question are characteristics that make it possible

to classify a product (as belonging or not belonging to a certain class, being or not being admissible, etc.) on the basis of a minimum number of measurements or tests. In a medical context, one is looking for a minimum number of tests that one has to perform in order to safely diagnose a disease, or, which is the same thing, be able to distinguish between diseases showing similar symptoms. Other applications abound, and the problem can also be formulated somewhat more generally by assigning weights to the properties and asking for a minimum-weight (rather than a minimum-cardinality) set of properties to satisfy the required condition. Here the interpretation of the weights may be the cost of the tests or measurements (in industry), the risk involved in the tests (in medicine), etc.

The formulation of this problem as a set covering problem is not so straightforward as in the other examples discussed above. Let $D = (d_{ij})$ be the incidence matrix of objects versus properties, i.e., let D have a row for every object and a column for every property, with $d_{ij} = 1$ if object i has property j, $d_{ij} = 0$ otherwise. Then our problem can be stated as that of finding a minimum number of columns (or, if the properties are weighted, a minimum-weight subset of the column set) such that the submatrix of D consisting of these columns has no pair of rows that are componentwise equal; in other words, such that for every pair i, k of rows (objects), the submatrix in question contains at least one column (property) j such that $d_{ij} = 1$ and $d_{kj} = 0$, or $d_{ij} = 0$ and $d_{kj} = 1$.

Now define a new 0-1 matrix $A = (a_{ij})$ with $n = p$ columns, one for every column of D, and $m = \frac{1}{2} q(q-1)$ rows, one for every distinct pair of rows of D and such that, if row k of A corresponds to the pair of rows i_k, j_k of D, then

$$a_{k\ell} = \begin{cases} 1 & \text{if } d_{i_k \ell} \neq d_{j_k \ell} \\ 0 & \text{if } d_{i_k \ell} = d_{j_k \ell} \end{cases}$$

With this definition, the minimum test set problem can be formulated as the set covering problem

(SC) $\min\{cx \mid Ax \geq e,\ x_j = 0 \text{ or } 1,\ j \in N\}$,

where e and N are as before, while c_j is the weight assigned to property j (i.e., if we are solving the unweighted problem, $c_j = 1,\ \forall\ j \in N$).

3. Solution Methods

In this section we discuss a class of algorithms for solving set covering problems, based on cutting planes from conditional bounds [1, 2]. Several versions of such an algorithm were implemented jointly with A. Ho [3], and extensively tested on randomly generated and real world problems, with the conclusion that this algorithm is a reliable and efficient tool for solving large, sparse set covering problems of the kind that frequently occurs in practice. With a time limit of 10 minutes on a DEC 20/50, we have solved all but one of a set of 50 randomly generated set covering problems with up to 200 constraints, 2000 variables and 8000 nonzero matrix entries (here "solving" means finding an optimal solution and proving its optimality), never generating a branch and bound tree with more than 50 nodes. For problems that are too large to be solved within a reasonable time limit, the procedure usually finds good feasible solutions, with a bound on the distance from the optimum (for the one unsolved problem, this bound was 2.3%).

We consider the set covering problem (SC) introduced in section 1, and denote

$$M_j = \{i \in M | a_{ij} = 1\},\ j \in N; \qquad N_i = \{j \in N | a_{ij} = 1\},\ i \in M.$$

We also use the pair of dual linear programs

(L) $\quad \min \{cx | Ax \geq e,\ x \geq 0\}$

and

(D) $\quad \max \{ue | uA \leq c,\ u \geq 0\}$

associated with (SC).

A 0-1 vector x satisfying $Ax \geq e$ is called a <u>cover</u>, and $S(x) = \{j \in N | x_j = 1\}$ its <u>support</u>. A cover whose support is nonredundant is <u>prime</u>. For a cover x, we denote $T(x) = \{i \in M | a^i x = 1\}$, where a^i is the i-th row of A.

The theory underlying the family of cutting planes from conditional bounds can be summarized as follows (for proofs of these statements, interpretation of the cuts in terms of conditional bounds, and further elaboration on their properties, see [2]).

Let z_U be some upper bound on the value of (SC), and let u be any feasible solution to (D), with $s = c - uA$, such that the condition

(1) $\quad \sum_{j \in S} s_j \geq z_U - ue$

is satisfied for some $S \subseteq N$. Let $S = \{j(1), \ldots, j(p)\}$, and let Q_i, $i = 1, \ldots, p$, be any collection of subsets of N satisfying

(2) $\quad \sum_{i | j \in Q_i} s_{j(i)} \leq s_j, \qquad j \in N.$

Then every cover x such that $cx < z_U$ satisfies the disjunction

(3) $\quad \bigvee_{i=1}^{p} (x_j = 0,\ j \in Q_i).$

Further, for any choice of indices $h(i) \in M$, $i = 1,\ldots, p$, the disjunction (3) implies the inequality

(4) $$\sum_{j \in W} x_j \geq 1$$

where

$$W = \bigcup_{t=1}^{p} (N_{j(i)} \setminus Q_i).$$

Finally, if $j(i) \in Q_i$, $i = 1,\ldots,p$, and if $\bar{x}$ is a cover such that $S \subseteq S(\bar{x})$, and $h(i) \in T(\bar{x}) \cap M_{j(i)}$, $i = 1,\ldots,p$, then the inequality (4) cuts off $\bar{x}$ and defines a facet of

$$P = \text{conv}\{x \in R^n | Ax \geq e, \sum_{j \in W} x_j \geq 1, x \geq 0, x_j \text{ integer}, j \in N\},$$

where conv V means the convex hull of the set V.

Using the above results, one can generate a sequence of cutting planes that are all distinct from each other, by generating a sequence of covers x and feasible solutions u to (D). The covers x provide upper bounds, while the vectors u provide lower bounds on the value of (SC). Since every inequality that is generated cuts off a cover satisfying all previously generated inequalities, and the number of distinct covers is finite, the procedure ends in a finite number of iterations, with an optimal cover at hand.

The algorithm alternates between two sets of heuristics, one of which finds a "good" prime cover x for the current problem and a (possibly improved) upper bound, while the other generates a feasible solution to (D) satisfying condition (1) for $S = S(x)$, and from it a cutting plane (4) that cuts off x, as well as a (possibly improved) lower bound. Whenever a disjunction (3) is obtained with $p = 1$, all the variables indexed by Q_1 are set to 0. The second set of heuristics is periodically supplemented by subgradient optimization to obtain sharper lower bounds.

Though this procedure in itself is guaranteed to find an optimal cover in a finite number of iterations, for large problems this may take too many cuts. Therefore, as soon as the rate of improvement in the bounds decreases beyond a certain value, the algorithm branches.

A schematic flowchart of the algorithm is shown in Fig. 1. PRIMAL designates the set of heuristics used for finding prime covers, DUAL the heuristics used for finding feasible dual solutions. TEST is the routine for fixing variables at 0. CUT generates a cutting plane violated by the current cover. SGRAD uses subgradient optimization in an attempt to find an improved dual solution and lower bound. BRANCH is the branching routine which breaks up the current problem into a number of subproblems, while SELECT chooses a new subproblem to be processed.

The four decision boxes of the flowchart can be described as follows. Let z_U and z_L be the current upper bound and lower bound, respectively, on the value of (SC).

1. If $z_L \geq z_U$, the current subproblem is fathomed (1.1). If $z_L < z_U$ and some variable belonging to the last prime cover has been fixed at 0, a new cover has to be found (1.2). Otherwise, a cut is generated (1.3).

2. After adding a cut, the algorithm returns to PRIMAL (2.1) unless the iteration counter is a multiple of some constant α, in which case (2.2) it uses subgradient optimization in an attempt to improve upon z_L. Based on some experimentation, the value of α is chosen such that $(|M|/10) \leq \alpha \leq (|M|/20)$.

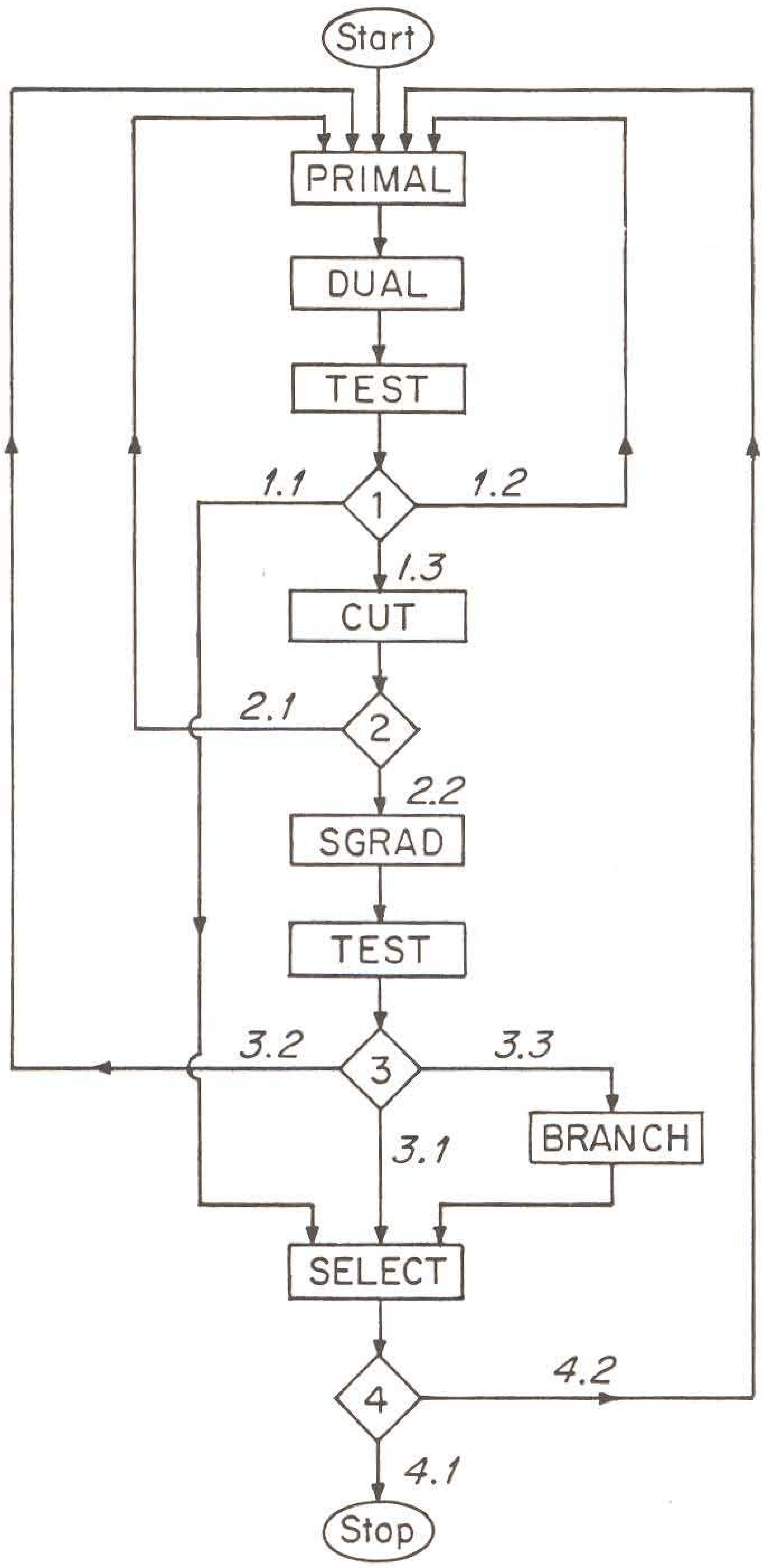
Start
PRIMAL
DUAL
TEST
1.1
1
1.2
1.3
CUT
2.1
2
2.2
SGRAD
TEST
3.2
3
3.3
BRANCH
3.1
SELECT
4
4.2
4.1
Stop

Fig. 1

3. If $z_L \geq z_U$, the current subproblem is fathomed (3.1). If $z_L < z_U$ but the gap $z_U - z_L$ has decreased by at least $\varepsilon > 0$ during the last β iterations for some constant β, we continue the iterative process (3.2). Otherwise, we branch (3.3). Again, based on some experimentation, we use $\varepsilon = 0.5$ and $\beta = 4\alpha$, with α as defined in 2.

4. If there are no active subproblems, the algorithm stops: the cover associated with z_U is optimal (4.1). Otherwise, it applies the iterative procedure to the selected subproblem (4.2).

Next we briefly discuss the various ingredients of the algorithm and their role in making the procedure efficient.

Primal heuristics. Most of the procedures we use to generate prime covers are of the "greedy" type, in that they construct a cover by a sequence of steps, each of which consists of the selection of a variable x_j that minimizes a certain function of the coefficients of x_j. They differ in the function f used to evaluate the variables. If k_j denotes the number of positive coefficients of x_j in those rows of the current constraint set not yet covered, the general form of the evaluation function is $f(c_j, k_j)$.

Since it is computationally cheaper to consider only a subset of variables at a time and since every row must be covered anyhow, we restrict the choice at each step to those variables having a positive coefficient in some specified row $i_* \in M$, where M indexes the rows. Denoting by R the set of rows not yet covered and by S the support of the cover to be constructed, the basic procedure that we use can be stated as follows.

Step 0. Set $R = M$, $S = \emptyset$, $t = 1$, and go to 1.

Step 1. If $R = \emptyset$, go to 2. Otherwise let $k_j = |M_j \cap R|$, choose $i_* \in R$, and choose $j(t)$ such that

$$f(c_{j(t)}, k_{j(t)}) = \min_{j \in N_{i_*}} f(c_j, k_j).$$

Set $R \leftarrow R \setminus M_{j(t)}$, $S \leftarrow S \cup \{j(t)\}$, $t \leftarrow t + 1$, and go to 1.

Step 2. Consider the elements $i \in S$ in order, and if $S \setminus \{i\}$ is the support of a cover, set $S \leftarrow S \setminus \{i\}$. When all $i \in S$ have been considered, S defines a prime cover.

As to the choice of i_* in Step 1, we order the rows of the initial coefficient matrix once and for all according to decreasing N_i, and then always choose i_* as the last element of the ordered set R. Since the cuts generated in the procedure also tend to have a decreasing number of 1's, i.e. later cuts tend to have fewer positive coefficients than earlier cuts, this rule approximates the criterion of always choosing a row with a minimum number of positive coefficients.

If the set N_{i_*} in step 1 is replaced by N and step 2 is removed, i.e. if the choice of columns is not restricted every time to a particular row and the procedure is allowed to stop whenever a cover is obtained, whether prime or not, then the above procedure is the greedy heuristic shown by Chvátal [4] to have the following property: if z_{opt} is the value of (SC) and z_{heu} the value of the solution found by the heuristic, then

$$z_{heu} / z_{opt} \leq \sum_{j=1}^{d} \frac{1}{j},$$

where

$$d = \max_{j \in N} |M_j|,$$

and this bound is best possible. From a practical standpoint, this bound is of course very poor and it was shown by Ho [6] that there is no better bound for any function f used in the above procedure. Ho's proof of this result relies on the construction of examples for which the worst case bound is attained, and different families of functions f require different examples. This suggests as a practical remedy against the poor worst case performance of the heuristic, the intermittent use of several functions f rather than a single one. This idea was implemented and tested with reasonably good results. The following five functions were considered: (i) c_j; (ii) c_j/k_j; (iii) $c_j/\log_2 k_j$; (iv) $c_j/k_j \log_2 k_j$; (v) $c_j/k_j \ln k_j$. In cases (iii) and (iv), $\log_2 k_j$ is to be replaced by 1 when $k_j = 1$; and in case (v), $\ln k_j$ is to be replaced by 1 when $k_j = 1$ or 2.

The five functions were tested on a set of randomly generated problems, with the result that mixing them intermittently rather than using any one of them by itself improves the quality of the solution considerably.

A different primal heuristic, that we use every time the subgradient method is applied to obtain an improved lower bound, is based on the reduced costs $s_j = c_j - ua_j$ produced by the subgradient method. This procedure sets $x_j = 1$ if $s_j = 0$, $x_j = 0$ otherwise. The resulting vector x either is a cover, or else if row i is uncovered, then $s_j > 0$ for all $j \in N_i$, and u_i can be increased to $u_i + \min_{j \in N_i} s_j$. This creates at least one new reduced cost s_k equal to 0, and for each such k we set $x_k = 1$. We proceed this way until every row is covered, after which we apply step 2 of the first heuristic to make the cover prime. This second heuristic, though considerably more expensive than the first one (because of the computational effort involved in the subgradient method), consistently outperformed the first heuristic.

Dual heuristics and subgradient optimization. The purpose of these procedures is to find, at a low computational cost, "good" feasible solutions to (D), hence "good" lower bounds are the value of (SC). The heuristics used are again of the greedy type, in that they construct a feasible solution to (D) by a sequence of steps, each of which consists of selecting a row i_* with a small number of positive coefficients, and assigning to u_{i_*} the maximum value that can be assigned without violating the constraints or changing some earlier value assignment. In choosing i_*, priority is given to $i \in T(x) = \{i \in M | a^i x = 1\}$, where x is the current cover. This is done in order to obtain a reduced cost vector s = c - uA that satisfies condition (1) for S = S(x), since it is known (see [2]) that this is the case if u satisfies u(Ax-e) = 0.

While this heuristic (used with minor variations depending on the situation) provides reasonably good solutions to (D) at a very low computational cost, a sharper lower bound could of course be obtained by solving (D) to optimality. After sufficient cuts have been added, the value z_L of (D) may exceed z_U, thus bringing the procedure to an end. However, the computational effort involved in repeatedly solving (D) by the simplex method is considerable, and increases about quadratically with the number of cuts added to the constraint set of (SC). On the other hand, one can use subgradient optimization to find a near-optimal solution to (D) at a computational cost that increases only linearly with the number of cuts added. This is what we are doing periodically in order to generate lower bounds stronger than those provided by the heuristic.

Our experience with the subgradient method has been that although it is more expensive than the dual heuristics often by 1 or 2 orders of magnitude, it nevertheless pays off if used sparingly, in combination with the heuristics.

On the one hand, it usually improves the lower bound; on the other, it produces a set of reduced costs that can be used to obtain improved covers, as explained in connection with the primal heuristics. At the same time, subgradient optimization cannot replace the dual heuristics, since it usually provides fractional solutions to (D) and such solutions tend to yield weaker cuts than the integer solutions obtained by the heuristic.

Fixing variables and generating cuts. Every time a new solution u to (D) is obtained either by the heuristic or by subgradient optimization, the algorithm searches for variables x_j such that $s_j \geq z_U - ue$, and fixes them at 0. This feature comes into play from early on in the procedure, and in the randomly generated test problems that we solved, the number of variables left by the time the first branching occurred, was almost always close to the initial number m of constraints.

To generate cuts, the algorithm uses the results stated at the beginning of this section. In order to obtain a cut (4) as strong as possible, i.e. with $|W|$ as small as possible, the construction of the sets Q_i and the choice of the indices $h(i)\epsilon M$ is done sequentially, so that at each step the set $N_{h(i)}\backslash Q_i$ is minimized. The cut generating subroutine is as follows. Let x be a cover with S(x) and T(x) defined as before, let u be a feasible solution to (D), with s = c-uA, and assume that s satisfies (1) for S = S(x).

Step 0. Set $W = \emptyset$, $S = \{j\epsilon S(x)|s_j > 0\}$, y = ue, t = 1, and go to 1.

Step 1. Let

$$v_t = \min\{\max_{j\in S} s_j, \min_{j\in S}\{s_j | s_j \geq z_U - y\}\},$$

$$J = \{j\in S | s_j = v_t\}, \qquad Q = \{j\in N | s_j \geq v_t\}, \qquad M_J = \bigcup_{j\in J} M_j .$$

Choose i(t) such that

$$|N_{i(t)} \setminus Q\cup W| = \min_{i\in T(x)\cap M_J} |N_i \setminus Q\cup W|$$

and let $\{j(t)\} = J\cap N_{i(t)}$.

Then set $W \leftarrow W\cup(N_{i(t)}\setminus Q)$, $y \leftarrow y + s_{j(t)}$. If $y \geq z_U$, go to 2.

Otherwise set $S \leftarrow S\setminus \{j(t)\}$,

$$s_j \leftarrow \begin{cases} s_j - s_{j(t)} , & j\in Q\cap N_{i(t)} \\ s_j , & \text{otherwise} \end{cases}$$

$t \leftarrow t + 1$, and go to 1.

Step 2. Add to (SC) the inequality

$$\sum_{j\in W} x_j \geq 1.$$

This procedure terminates after a number of iterations equal to the number of $j\in S(x)$ such that $s_j > 0$, with an inequality satisfied by every cover better than the one associated with z_U, and violated by the cover x.

Typically, the cuts tend to become successively stronger during the procedure, the last few cuts often having just one or two 1's. The total number of cuts required to solve an m x n problem tends to increase with both m and n. For the randomly generated sparse problems solved in our experiment, the number of cuts needed was typically less than 3m or n/3. This of course refers to the number of cuts required when the cuts are used within the framework of an algorithm that also uses implicit enumeration. The cuts by themselves, without branching, were able to solve all 20 test problems from the literature that we could obtain, and all but one of 10 randomly generated test problems with m = 100 and n = 100, 200,...,1000. As to the larger problems, six of the ten 200 X 1000 problems and four of the ten 200 x 2000 problems that we generated, were solved by cutting planes only without branching.

Branching and node selection. We branch whenever the gap $z_U - z_L$ decreases by less than $\varepsilon = 0.5$ during a sequence of 4α iterations, where α is the frequency of applying the subgradient method (in number of iterations). The algorithm uses two branching rules intermittently. The first one is based on disjunction (3), the second one is a variant of the dichotomy proposed by Etcheberry [5]. Since our tests showed that none of the two rules dominates the other, we use both rules, with the following choice criterion: since rule 1 fixes more variables, but at the cost of creating more branches, we prefer rule 1 only if it fixes more variables than could be fixed by creating the same number of branches through binary (dichotomic) branching. More precisely, we choose rule 1 if, while creating p branches, it fixes at least $p \log_2 p$ variables. As to node selection, we use the LIFO rule.

Computational experience. A detailed account of our computational experience is to be found in [3]. Here we reproduce only the results on the largest 10 test problems, a set of randomly generated problems with 200 constraints and 2,000 variables, with 8,000 non-zero entries in the coefficient matrix and with costs drawn from the interval [1,100]. The results are shown in Table 1.

Table 1. Results on 10 randomly generated problems.

No.	z_{opt}	Before first branching z_U	z_L	Variables left	Nodes in search tree	Cuts	Time Dec 20/50 seconds
1	253	256	250.6	204	30	473	327.9
2*	307**	315	299.3	408	>51	>625	>600
3	226	226	226.0	0	1	0	26.9
4	242	247	240.3	258	49	765	393.2
5	211	211	211.0	0	1	15	38.7
6	213	213	213.0	0	1	10	32.7
7	293	296	291.0	173	15	298	248.7
8	288	288	286.1	125	28	413	241.4
9	279	281	276.2	181	7	118	140.6
10	265	265	265.0	0	1	0	25.9

* Time limit of 10 minutes exceeded.

** Best solution found in 10 minutes.

Based on our computational experience, we can assert that the above described algorithm performs considerably better than earlier procedures proposed in the literature, and is in fact a reasonably reliable, efficient tool for solving large, sparse set covering problems, as well as for finding good approximate solutions to problems that are too hard to solve exactly.

References

[1] E. Balas, "Set Covering with Cutting Planes from Conditional Bounds." In A. Prékopa (editor), Survey of Mathematical Programming (Proceedings of the International Symposium on Mathematical Programming, Budapest, August 1976), Hungarian Academy of Sciences, 1980, p. 393-422).

[2] E. Balas, "Cutting Planes from Conditional Bounds: A New Approach to Set Covering". Mathematical Programming Study 12, 1980, p. 19-36.

[3] E. Balas, and A. Ho, "Set Covering Algorithims Using Cutting Planes, Heuristics and Subgradient Optimization". Mathematical Programming Study 12, 1980, p. 37-60.

[4] V. Chvátal, "A Greedy Heuristic for the Set Coverings Problem". Mathematics of Operations Research, 1980, p. 233-236.

[5] J. Etcheberry, "The Set Covering Problem: A New Implicit Enumeration Algorithm." Operations Research, 23, 1977, p. 760-772.

[6] A. Ho, "Worst Cast Analysis of a Class of Set Covering Heuristics". GSIA, Carnegie-Mellon University. June 1979 (to appear in Mathematical Programming).

Appendix I

Bibliography for Set Covering and Set Partitioning: Theory and Algorithms

Andrew, G., Hoffmann, Th. and Krabek, Ch. [1968]: "On the Generalized Set Covering Problem." CDC, Data Centers Division, Minneapolis

Balas, E. [1964]: "Un algorithme additif pour la résolution des programmes linéaires a variables bivalentes." C.R. Acad. Sci. Paris, 258, 3817-3820.

Balas, E. [1965]: "An Additive Algorithm for Solving Linear Programs With Zero-One Variables." Operations Research, 13, 517-546.

Balas, E. [1974a]: "Intersection Cuts from Disjunctive Constraints." MSRR No. 330, Carnegie-Mellon University, Pittsburgh, February.

Balas, E. [1974b]: "Disjunctive Programming: Properties of the Convex Hull of Feasible Points." MSRR No. 348, Carnegie-Mellon University, Pittsburgh, July.

Balas, E. [1975a]: "Facets of the Knapsack Polytope." Mathematical Programming, 8, 146-164.

Balas, E. [1975b]: "Disjunctive Programming: Cutting Planes from Logical Conditions." Nonlinear Programming 2, O. L. Mangasarian, R. R. Meyer and S. M. Robinson, eds., Academic Press, New York, 279-312.

Balas, E. [1975c]: "Some Valid Inequalities for the Set Partitioning Problem." MSRR No. 368, Carnegie-Mellon University, Pittsburgh, July (revised January 1976).

Balas, E. [1976]: "A Disjunctive Cut for Set Partitioning." W.P. 57-75-76, Carnegie-Mellon University, Pittsburgh, January.

Balas, E. [1979]: "Disjunctive Programming." Annals of Discrete Mathematics, 5, 3-51.

Balas, E. [1979]: "Set Covering with Cutting Planes from Conditional Bounds," in A. Prekopa (editor), Survey of Mathematical Programming, Hungarian Academy of Sciences, Budapest, 393-422.

Balas, E. [1980]: "Cutting Planes from Conditional Bounds: A New Approach to Set Covering." Mathematical Programming Study 12, 19-36.

Balas, E., Gerritsen, R. and Padberg, M. W. [1974]: "An All-Binary Column Generating Algorithm for Set Partitioning." Presented at the ORSA-TIMS meeting in Boston, April.

Balas, E. and Ho, A. [1980]: "Set Covering Algorithms Using Cutting Planes, Heuristics and Subgradient Optimization: A Computational Study." Mathematical Programming Study 12, 37-60.

Balas, E. and Jeroslow, R. G. [1975]: "Strengthening Cuts for Mixed Integer Programs." MSRR No. 359, Carnegie-Mellon University, Pittsburgh, February.

Balas, E. and Padberg, M. W. [1972a]: "On the Set Covering Problem." Operations Research, 20, 1152-1161.

Balas, E. and Padberg, M. W. [1973]: "Adjacent Vertices of the Convex Hull of Feasible 0-1 Points." MSRR No. 298, Carnegie-Mellon University, Pittsburgh, November-April.

Balas, E. and Padberg, M. W. [1975]: "On the Set Covering Problem, II. An Algorithm for Set Partitioning." Operations Research, 23, 74-90.

Balas, E. and Padberg, M. W. [1976]: "Set Partitioning: A Survey." SIAM Review, 18, 1976, p. 710-760. Reprinted in N. Christofides (editor), Combinatorial Optimization, J. Wiley, 1979.

Balas, E. and Samuelsson, H. [1973]: "Finding a Minimum Node Cover in an Arbitrary Graph." MSRR No. 325, Carnegie-Mellon University, Pittsburgh, November.

Balas, E. and Zemel, E. [1974]: "Facets of the Knapsack Polytope from Minimal Covers." MSRR No. 352, Carnegie-Mellon University, Pittsburgh, December.

Balas, E. and Zemel, E. [1975]: "All the Facets of 0-1 Programming Polytopes with Positive Coefficients." MSRR No. 374, Carnegie-Mellon University, Pittsburgh, October.

Balas, E. and Zemel, E. [1976a]: "Graph Substitution and Set Packing Polytopes." MSRR No. 384, Carnegie-Mellon University, Pittsburgh, January.

Balas, E. and Zemel, E. [1976b]: "Critical Cutsets of Graphs and Canonical Facets of Set Packing Polytopes." MSRR No. 385, Carnegie-Mellon University, Pittsburgh, February.

Balinski, M. L. [1969]: "Labeling to Obtain a Maximum Matching." Combinatorial Mathematics and Its Applications, R. C. Bose and T. A. Dowling, eds., University of North Carolina Press, Charlottesville, 585-601.

Balinski, M. L. [1970]: "On Maximum Matching, Minimum Covering and Their Connections," Proc. Princeton Symposium on Mathematical Programming, H. W. Kuhn, ed., Princeton University Press, Princeton, N.J.

Balinski, M. L. [1972]: "Establishing the Matching Polytope." Journal of Combinatorial Theory, 13, 1-13.

Bellmore, M. and Ratliff, A. D. [1971]: "Set Covering and Involutory Bases." Management Science 18, 194-206.

Berge, C. [1957]: "Two Theorems in Graph Theory." Proc. Nat. Acad. of Sciences. USA, 43, 842-844.

Berge, C. [1961]: "Färbung von Graphen, deren säntliche bzw. deren ungerade Kreise starr sind [Zusammensassung]." Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe, 114.

Berge, C. [1970]: Graphes et Hypergraphes, Dunod, Paris; English transl., North-Holland, Amsterdam, 1973.

Berge, C. [1972]: "Balanced Matrices." Mathematical Programming, 2, 19-31.

Christofides, N. and Korman, S. [1973]: "A Computational Survey of Methods for the Set Covering Problem." Rep. 73/2, Imperial College of Science and Technology, London, April.

Christofides, N. [1975]: Graph Theory: An Algorithmic Approach. Academic Press, New York.

Chvátal, V. [1972]: "On Certain Polytopes Associated with Graphs." CRM-238, Université de Montréal, Montréal, October; J. Comb. Theory, to appear.

Chvátal, V. [1978]: "A Greedy Heuristic for the Set Covering Problem." Publication 284, Departement d'Informatique, et de Recherche Operationnelle; Université de Montréal.

Dantzig, G. B. [1963]: Linear Programming and Extensions. Princeton University Press, Princeton, N.J.

Délorme, J. [1974]: "Contribution à la résolution du problème de recouvrement: méthodes de troncatures." Thèse de Docteur Ingénieur, Université Paris VI, Paris.

Délorme, J. and Heurgon, E. [1975]: Problèmes de partitionnement: exploration arborescente ou méthodes de troncatures?" Rev. Francaise Autom., Inf. Rech. Oper., 9, V-2, 53-65.

Edmonds, J. [1962]: "Covers and Packings in a Family of Sets." Bull. Amer. Math. Soc., 68, 494-499.

Edmonds, J. [1965a]: "Maximum Matching and a Polyhedron with 0, 1 Vertices." J. of Res. Nat. Bur. of Standards, 69B, 125-130.

Edmonds, J. [1965b]: "Paths, Trees and Flowers." Canad. J. Math., 17, 449-467.

Etcheberry, J. [1977]: "The Set Covering Problem: A New Implicit Enumeration Algorithm." Operations Research, 25, 760-772.

Fréhel, J. [1974]: Problèmes de partition: algorithme du simplexe, expériences numériques." Rev. Francaise Auto. Inf. Rech. Oper., to appear.

Fulkerson, D. R. [1971]: "Blocking and Anti-Blocking Pairs of Polyhedra." Mathematical Programming, 1, 168-194.

Fulkerson, D. R. [1973]: "On the Perfect Graph Theorem," Mathematical Programming, T. C. Hu and S. M. Robinson, eds., Academic Press, New York.

Gallai, T. [1958]: "Über Extreme Punkt-und Kantenmengen." Ann. Univ. Sci Budapest, Eötvös, Sect. Math., 2, 133-138.

Garfinkel, R. S. and Nemhauser, G. L. [1969]: "The Set Partitioning Problem: Set Covering with Equality Constraints." Operations Research, 17, 848-856.

Garfinkel, R. S. and Nemhauser, G. L. [1972]: Integer Programming, John Wiley, New York.

Garfinkel, R. S. and Nemhauser, G. L. [1972]: "Optimal Set Covering: A Survey," in A. M. Geoffrion (editor), Perspectives on Optimization, Addison-Wesley, 164-183.

Geoffrion, A. M. [1967]: "Integer Programming by Implicit Enumeration and Balas' Method," SIAM Review, 7, 178-190.

Glover, F. [1965]: "A Multiphase Dual Algorithm for the Zero-One Integer Programming Problem." Operations Research, 13, 94-120.

Glover, F. [1975]: "Polyhedral Annexation in Mixed Integer Programming." Mathematical Programming, 9.

Glover, F. and Klingman, D. [1973a]: "The Generalized Lattice Point Problem." Operations Research, 21, 141-156.

Glover, F. and Klingman, D. [1973b]: "Improved Convexity Cuts for Lattice Point Problems." CS133, University of Texas, Austin, April.

Gomory, R. E. [1963a]: "An Algorithm for Integer Solutions to Linear Programs." Recent Advances in Mathematical Programming, R. L. Graves and P. Wolfe, eds., McGraw-Hill, New York.

Gomory, R. E. [1963b]: "All-Integer Integer Programming Algorithm." Industrial Scheduling, J. F. Muth and G. L. Thompson, eds., Addison-Wesley, Reading, Mass.

Gondran, M. [1973]: "Un outil pour la programmation en nombres entiers: la méthode des congruences décroissantes." Rev. Francaise Auto. Inf. Rech. Oper., 3, 35-54.

Gondran, M. and Laurière, J. L. [1974]: "Un algorithme pour le problème de partitionnement." Rev. Francaise Auto. Inf. Rech. Oper., 8, 27-70.

Graves, G. W. and Whinston, A. B. [1968]: "A New Approach to Discrete Mathematical Programming." Management Science, 15, 177-190.

Hammer, P. L., Johnson, E. L. and Peled, U. N. [1975]: "Facets of Regular 0-1 Polytopes." Mathematical Programming, 8, 179-206.

Hammer, P. L., Johnson, E. L. and Peled, U. N. [1974]: "The Role of Master Polytopes in the Unit Cube." CORR 74-25, University of Waterloo, Waterloo, Ont., Canada, October.

Harary, F. [1969]: Graph Theory, Addison-Wesley, Reading, Mass.

Held, M. and Karp, R. M. [1971]: "The Traveling-Salesman Problem and Minimum Spanning Trees: Part II." Mathematical Programming, 1, 6-25.

Held, M., Wolfe, P., and Crowder, H. D. [1974]: "Validation of Sub-gradient Optimization." Mathematical Programming, 6, 62-88.

Heurgon, E. [1972]: "Un problème de recouvrement exact: l'habillage des horaires d'une ligne d'autobus. Rev. Francaise Auto. Inf. Rech. Oper., 6.

House, R., Nelson, L. and Rado, T. [1965]: "Computer Studies of a Certain Class of Linear Integer Programs," in Lavi and Vogel (editors), Recent Advances in Optimization Techniques, Wiley, New York.

Jeroslow, R. G. [1974]: "Principles of Cutting Plane Theory: Part I." Carnegie-Mellon University, Pittsburgh, February.

Johnson, E. L. [1974]: "A Class of Facets of the Master 0-1 Knapsack Polytope." Watson Research Center, IBM, Yorktown Heights, N.Y., November.

Lemke, C. E., Salkin, H. M. and Spielberg, K., [1971]: "Set Covering by Single Branch Enumeration with Linear Programming Subproblems." Operations Research, 19, 998-1022.

Lovász, L. [1972]: "Normal Hypergraphs and the Perfect Graph Conjecture." Discrete Mathematics, 2, 253-267.

Lovász, L. [1975]: "On the Ratio of Optimal Integral and Fractional Covers." Discrete Mathematics, 13, 383-390.

Marsten, R. E. [1974]: "An Algorithm for Large Set Partitioning Problems." Management Science, 20, 779-787.

Martin, G. T. [1963]: "An Accelerated Euclidean Algorithm for Integer Programming." Recent Advances in Mathematical Programming, R. L. Graves and P. Wolfe, eds., John Wiley, New York.

Martin, G. T. [1969]: "Gomory Plus Ten." Presented at the ORSA Meeting in Miami, November.

Michaud, P. [1972]: "Exact Implicit Enumeration Method for Solving the Set Partitioning Problem. IBM J. Res. Develop., 16, 573-578.

Ming-Te Lu [1970]: "A Computerized Airline Crew Scheduling System." Ph.D. Thesis, University of Minnesota, Minneapolis.

Nemhauser, G. L. and Trotter, L. E. [1974]: "Properties of Vertex Packing and Independence System Polyhedra." Mathematical Programming, 6, 48-61.

Nemhauser, G. L., Trotter, L. E. and Nauss, R. M. [1972]: "Set Partitioning and Chain Decomposition." Technical Report No. 161, Cornell University, Ithaca, N.Y.

Owen, G. [1973]: "Cutting Planes for Programs with Disjunctive Constraints." J. Optimization Theory and Appl., 11, 49-55.

Padberg, M. W. [1971]: "Essays in Integer Programming." Ph.D. Thesis, Carnegie-Mellon University, Pittsburgh, May.

Padberg, M. W. [1973a]: "On the Facial Structure of Set Packing Polyhedra." Mathematical Programming, 5, 199-215.

Padberg, M. W. [1973b]: " A Note on Zero-One Programming." Operations Research, 23, p. 833-837.

Padberg, M. W. [1974]: "Perfect Zero-One Matrices." Mathematical Programming, 6, 180-196.

Padberg, M. W. [1974b]: "Perfect Zero-One Matrices - II," Proceedings in Operations Research, 3, Physica-Verlag, Wurzburg-Wien, 75-83.

Padberg, M. W. [1974c]: "Characterizations of Totally Unimodular, Balanced and Perfect Matrices." Combinatorial Programming: Methods and Applications, B. Roy, ed., D. Reidel Pub., Dordrecht/Boston, 275-284.

Padberg, M. W. [1975a]: "Almost Integral Polyhedra Related to Certain Combinatorial Optimization Problems." Working paper 75-25, GBA, New York Univ., New York, Linear Algebra and Appl., to appear.

Padberg, M. W. [1975b]: "On the Complexity of Set Packing Polyhedra." Working paper No. 75-105, GBA, New York University, New York.

Padberg, M. W. and Rao, M. R. [1973]: "The Travelling Salesman Problem and a Class of Polyhedra of Diameter Two." IIM Preprint No. I/73-5, International Institute of Management, Berlin; Math. Programming, to appear.

Pierce, J. F. [1968]: "Application of Combinational Programming to a Class of All Zero-One Integer Programming Problems. Management Science, 15, 191-209.

Pierce, J. F. and Lasky, J. S. [1973]:"Improved Combinational Programming Algorithms for a Class of All Zero-One Integer Programming Problems." Management Science, 19, 528-543.

Pollatschek, M. A. [1970]: "Algorithms on Finite Weighted Graphs." Ph.D. thesis, Technion, Haifa (in Hebrew, with an English summary).

Pulleyblank, W. and Edmonds, J. [1973]: "Facets of 1-Matching Polyhedra," CORR, University of Waterloo, Waterloo, Ont., Canada, March

Roth, R. [1969]: "Computer Solutions to Minimum Cover Problems." Operations Research, 17, 455.

Roy, B. [1969]: Algèbre moderne et Théorie des Graphes, I, Dunod, Paris.

Roy, B. [1970]: Algèbre moderne et Théorie des Graphes, II, Dunod, Paris.

Salkin, H. M. and Koncal, R. D. [1973]: "Set Covering by an All Integer Algorithm: Computational Experience." J. Assoc. Comput. Math., 20, 189-193.

Samuelsson, H. [1974a]: "Integer Programming Duality for Set Packing-Partitioning Problems." MSRR No. 337, Carnegie-Mellon University, Pittsburgh, March.

Samuelsson, H. [1974b]: "A Symmetric Ascent Method for Finitely Regularizable Linear Programs." W.P.6-74-75, Carnegie-Mellon University, Pittsburgh, August.

Simmonard, M. [1966]: Linear Programming. Prentice-Hall, Englewood Cliffs, N.J.

Thiriez, H. M. [1971]: "The Set Covering Problem: A Group Theoretic Approach." Rev. Francaise Auto. Inf. Rech. Oper., 5, 83-104.

Trotter, L. E. [1973]: "Solution Characteristics and Algorithms for the Vertex Packing Problem." Technical Report No. 168, Cornell University, Ithaca, N.Y.

Trotter, L. E. [1974]: "A Class of Facet Producing Graphs for Vertex Packing Polyhedra." Technical Report No. 78, Yale University, New Haven, Conn., February.

Trubin, V. A. [1969]: "On a Method of Solution of Integer Linear Programming Problems of a Special Kind." Soviet Math. Dokl., 10, 1544-1596.

Weyl, H. [1935]: "Elementare Theorie der konvexen Polyeder." Comm. Math. Helv., 7, 290-306; English transl., in Contributions to the Theory of Games, vol. I, Annals of Mathematics Studies, No. 24, Princeton, N.J., 1950, 3-18.

Wolsey, L. A. [1975]: "Faces for Linear Inequalities in Zero-One Variables." Mathematical Programming, 8, 165-178.

Wolsey, L. A. [1974]: "Facets and Strong Valid Inequalities for Integer Programming." CORE, April.

Zemel, E. [1974]: "Lifting the Facets of 0-1 Polytopes." MSRR No. 354, Carnegie-Mellon University, Pittsburgh, December.

Zwart, P. B. [1972]: "Intersection Cuts for Separable Programming." School of Engrg. Appl. Sci., Washington Univ., St. Louis, Missouri, January.

Appendix II

Bibliography for Set Covering and Set Partitioning: Applications

(By area of application, chronologically within each area)

Crew Scheduling (Airline, Railroad, etc.)

1 Charnes, A., and M.H. Miller, "A Model for Optimal Programming of Railway Freight Train Movements", Management Sci., 3, 74-92 (1956).

2 McCloskey, J.F., and F. Hanssman, "An Analysis of Stewardess Requirements and Scheduling for a Major Airline", Naval Res. Log. Quart. 4, 183-192, (1957).

3 Evers, G.H.E., "Relevant Factors Around Crew - Utilization", KLM Rep, AGIFORS Symposium, Killarney, Scotland (1965).

4 Steiger, F., "Optimization of Swiss Air's Crew Scheduling by an Integer Linear Programming Model", Swiss Air O.R. SDK 3.3.911, (1965).

5 Kolner, T.N., "Some Highlights of a Scheduling Matrix Generator System", United Airlines, (1966).

6 Niederer, M., "Optimization of Swissair's Crew Scheduling by Heuristic Methods Using Integer Linear Programming Models", AGIFORS Symposium, Killarney, Scotland (1966).

7 Agard, J., "Monthly Assignment of Stewards", Air France, AGIFORS Symposium, Killarney, Scotland (1966).

8 Arabeyre, J.P., "Methods of Crew Scheduling", Air France, AGIFORS Symposium, 1966.

9 Moreland, J.A., "Scheduling of Airline Flight Crews", M.S. Thesis, Massachusetts Institute of Technology, Cambridge (1966).

10 Agard, J., J. P. Arabeyre, and J. Vautier, "Génération Automatique de Rotations d'équipages", Rev. Francaise Auto. Inf. Rech. Oper. I-6 107-117 (1967).

11 Steiger, F., and M. Niederer, "Scheduling Air Crews by Integer Programming", Presented at IFIP Congress, Edinburgh, Scotland (1968).

12 Thiriez, H.M., "Implicit Enumeration Applied to the Crew Scheduling Algorithm", Dept. of Aeronautics, Massachusetts Institute of Technology, Cambridge (1968).

13 Arabeyre, J.P., J. Fearnley, F. Steiger and W. Teather, "The Air Crew Scheduling Problem: A Survey", Trans. Sci. 3, 140-163, (1969).

14 Thiriez, H., "Airline Crew Scheduling- A Group Theoretic Approach", Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, October (1969).

15 Spitzer, M., "Solutions to the Crew Scheduling Problem", AGIFORS Symposium, October (1971).

Airline Fleet Scheduling

16 Levin, A., "Fleet Routing and Scheduling Problems for Air Transportation Systems," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, January (1969).

Truck Delivery

17 Dantzig, G.B., and J.H. Ramser, "The Truck Dispatching Problem", Management Sci. 6, 80-91, (1960).

18 Balinski, M.L., and M.H. Quandt, "On an Integer Program for a Delivery Problem," Operations Res., 12, 300-304 (1964).

19 Clarke, G., and S. W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", Operations Res., 12, 568-581 (1964).

20 Pierce, J.F., "Application of Combinatorical Programming Algorithms for a Class of All Zero One Integer Programming Problems", Management Sci., 15, 191-209, (1968).

Stock Cutting

21 Pierce, J.F., "Pattern Sequencing and Matching in Stock Cutting Operations", Tappi 53, 668-678 (1970).

Line and Capacity Balancing

22 Salveson, M.E., "The Assembly Line Balancing Problem", J. Indus. Eng., 6, 519-526 (1955).

23 Freeman, D.R., and J.V. Jucher, "The Line Balancing Problem," J. Indus. Eng., 18, 361-364 (1967).

24 Steinman, H., and R. Schwinn, "Computational Experience with a Zero-One Programming Problem", Operations Res., 17, 917-920 (1969).

Facility Location

25 Revelle, C., D. Marks, and J.C. Liebman, "An Analysis of Private and Public Sector Location Models", Management Sci., 16, 692-707 (1970).

26 Toregas, C., R. Swain, C. Revelle and L. Bergman, "The Location of Emergency Service Facilities", Operations Res., 19, 1363-1373 (1971).

26a Garfinkel, R. S., Neebe, A. W. and M. R. Rao, "The m-Center Problem: Bottleneck Facility Location," Working paper No. 7414, Graduate School of Management, University of Rochester, Rochester, N.Y. (1974).

Capital Investment

27 Valenta, J.R., "Capital Equipment Decisions: A Model for Optimal Systems Interfacing", M.S. Thesis, Massachusetts Institute of Technology, Cambridge, June (1969).

Switching Current Design and Symbolic Logic

28 Roth, J.P., "Algebraic Topological Methods for the Synthesis of Switching Systems - I", Trans. Amer. Math. Soc. 88, 301-326, (1950).

29 Quine, W.V., "A Way to Simplify Truth Functions", Amer. Math. Monthly, 62, 627-631, (1955).

30 McCluskey, E.J., Jr., "Minimization of Boolean Functions", Bell System Tech Journal 35, 1412-1444, (1956).

31 Petrick, S.R., "A Direct Determination of the Redundent Forms of a Boolean Function from the Set of Prime Implicants", AFCRC-TR-56-110, Air Force Cambridge Research Center, Cambridge, Mass. (1956).

32 Paul, M.C., and S.H. Unger, "Minimizing the Number of States in Incompletely Specified Sequential Functions", IRE Trans. on Electronic Computers Ec-8, 356-367, (1959).

33 Pyne, I.B., and E.J. McCluskey, Jr., "An Assay on Prime Implicant Tables", J. Soc. Indust. Appl. Math, 9, 604-631 (1961).

34 Cobham, A., R. Fridshal and J. H. North, "An Application of Linear Programming to the Minimization of Boolean Functions", Res. Rep. RC-472, IBM, Yorktown Heights, N.Y. (1961).

35 Cobham, A., R. Fridshal and J. H. North, "A Statistical Study of the Minimization of Boolean Functions Using Integer Programming", Res. Rep. R.C.-756, IBM, Yorktown Heights, N.Y. (1962).

36 Cobham, A., and J.H. North, "Extension of the Integer Programming Approach to the Minimization of Boolean Functions", Res. Rep. R.C. 915, IBM, Yorktown Heights, N.Y. (1963).

37 Root, J.C., "An Application of Symbolic Logic to a Selection Problem", Operations Res., 12, 519-526 (1964).

38 Balinski, M.L., "Integer Programming: Methods, Uses, Computation", Management Sci., 12, 253-313 (1965).

39 Gimpel, J.F., "A Reduction Technique for Prime Implicant Tables", IEEE Trans. on Electronic Computers, EC-14, 535-541, (1965).

Information Retrieval

40 Day, R.H., "On Optimal Extracting from a Multiple File Data Storage System: An Application of Integer Programming", Operations Res., 13, 482-494, (1965).

Marketing

41 Shanker, R. J., R. E. Turner and A. A. Zoltners: "Integrating the Criteria for Sales Force Allocation: A Set-Partitioning Approach." Working paper #48-72-3, Carnegie-Mellon Univ., Pittsburgh, December (1972).

Political Districting

42 Garfinkel, R.S., "Optimal Political Districting", Ph.D. Dissertation, Johns Hopkins Univ., Baltimore, Md. (1968).

43 Wagner, W.H., "An Application of Integer Programming to Legislative Redistricting", presented at the 34th National Meeting of ORSA (1968).

44 Garfinkel, R.S., and G.L. Nemhauser, Optimal Political Districting by Implicit Enumeration Techniques", Management Sci., 16, B495-B508 (1970).