TR\03\88 **July 1988**

A TREE SEARCH APPROACH FOR THE SOLUTION OF SET PROBLEMS USING ALTERNATIVE RELAXATIONS

by

Elia El—Darzi and Gautam Mitra

ABSTRACT

A number of alternative relaxations for the family of set problems (FSP) in general and set covering problems (SCP) in particular are introduced and discussed. These are (i) Network flow relaxation, (ii) Assignment relaxation, (iii) Shortest route relaxation, (iv) Minimum spanning tree relaxation. A unified tree search method is developed which makes use of these relaxations. Computational experience of processing a collection of test problems is reported.

Key words: integer programming, discrete optimisation, set covering, scheduling, assignment, branch and bound.

CONTENTS

1. Introductio to set problems(SP)

It has been known that integer programming can represent a wide class of discrete optimisation problems [DANT 63] and [WILL 85]. In this paper we are interested in a class of 0-1 integer programming problems: the set covering problem (SCP), the set partitioning problem (SPP) and together we refer to them as the set problems (SP). They are well known problems in the field of graph theory and combinatorial optimisation.

It is well accepted that these problems and their solution methods represent most successful instances of applying discrete models to solve industrial scheduling problems. Alternative algorithms based on graph theoretic approach for the SP have been investigated in this paper, using a collection of test problems which were put together to reflect real life applications.

The contents of this paper is organised as follows. In section 2 applications of the SP are briefly discussed and the collection of test problems is outlined. In section 3 a few heuristics which have become established way of preprocessing such problems prior to applying a full solution algorithms are described. The graph theoretic relaxations which we introduced constitute a new approach to solving SP and are described in section 4. The tree search algorithm is described in section 5 and the computational results are presented in section 6. An example illustrating the relaxations with a small SP model is set out in Appendix .

Notation and problem definition

Consider a set R={l,2,...,m} and a class H of subsets of R, such that H= {H_1, H_2,.....Hn}. Let J={l,2,...,n} be the set of indices for the subsets which make up the class H.

A cover for R is a subclass of H defined as {Hj|j $\in$ J_c} where j_c c j, satisfying

$\bigcup_{j \in Jc} H_j = R$. Let c_j be the cost of including H_j in the cover. Thus the minimum cost set covering problem is that of finding a cover $\{H_j \mid j \in J_c^*\}$ as above, such that

$$\sum_{j \in J_c^*} c_j \quad \text{is a minimum.}$$

The SCP may be also posed as a zero-one integer programming problem.

$$\text{Min} \quad \sum_{j=1}^{n} cjxj \tag{1.1}$$

subject to

$$\sum_{j=1}^{n} aijxj > 1, i = 1,....,m \tag{1.2}$$

$$x_j \in \{0,1\}, j = 1,....,n \tag{1.3}$$

where

$$x_j = \begin{cases} 1 \text{ if } H_j \text{ is included in the cover} \\ 0 \text{ otherwise} \end{cases}$$

and

$$aij = \begin{cases} 1 \text{ if } i \in H_j \\ 0 \text{ otherwise} \end{cases}$$

It is convenient to introduce the index sets R_i, i = l,2,...m, such that for a row i , R_i denotes the indices of the columns with unit entry. Similarly, the index sets Hj, j = l,2,...,n, denote the indices of the rows with unit entry in column j. Thus H_j and R_i are related to a_{ij}: as set out in (1.4) and (1.5)

$H_j = \{i1a_{ij} = 1, \ i=1.....m)$ for all j (1.4)

$$|H_j| = \sum_{i=1}^{m} a_{ij}$$

and

$R_i = \{j \mid a_{ij}=1, \ j=1,\ldots,n\}$ for all i

(1.5)

$$|R_i| = \sum_{j=1}^{n} a_{ij}.$$

A feasible solution to the SCP is called a cover. A prime cover x*, is a cover for which xj currently taking value one cannot be reduced to zero without violating a constraint. An optimal solution to the SCP is a prime cover if all the costs are positive.

The SPP may be defined as

$$\text{Min} \quad \sum_{j=1}^{n} c_j x_j \tag{1.6}$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j = 1, i = 1,....,m \tag{1.7}$$

$$x_j \in \{0,1\}, j = 1,....,n \tag{1.8}$$

and represents the minimum cost selection such that each member of R is included exactly once.

2.Applications of SP and test prodlems

It is known that SP represent a wide range of industrial scheduling and planning problems. These include bus crew scheduling [MTDD 85], air crew scheduling [BAFS 81], vehicle routing [CHRS 85], steiner problem [FNHT 74], facility location [DKST 81] and others. For a comprehensive survey on the application of the SP see Balas et al [BLPD 76] and Balas [BALS 83]. The many applications of the SP have constantly stimulated researchers to develop new algorithms for the SP. There are many algorithms which solve these problems. In testing performance of algorithms it is meaningful to use problem instances which are taken from real or (nearly) real applications. It is doubtful if randomly generated models have much value in testing algorithms which are designed to solve real problems. With this in mind we have collected a range of test problems taken from different contexts which are summarised below. For a full discussion of these models we

refer the readers to [ELDA 88] and [EDMT 88].

Summary of test problems

Table below gives a summary of test problems collected by us.

Problems names	Number of rows	Number of columns	Number of non zero entries	Density
Airline				
AIR1	158	416	1371	0.021
AIR2	153	1050	3510	0.022
AIR3	128	1100	3728	0.026
AIR4	148	1100	3670	0.022
AIR5	152	1043	3435	0.021
AIR6	139	1100	3731	0.024
Bus				
BUS1	28	231	628	0.097
BUS2	27	168	428	0.094
BUS3	55	1059	3227	0.055
BUS4	53	547	1401	0.048
BUS5	57	31	223	0.126
BUS6	108	245	1538	0.058
BUS7	213	2200	6090	0.013
BUS8	316	3015	12950	0.013

Problems names	Number of rows	Number of columns	Number of non zero entries	Density
Steiner				
STR1	117	27	351	0.111
STR2	330	45	990	0.067
Random				
RDM1	50	60	139	0.046
RDM2	51	61	138	0.044
RDM3	92	88	486	0.060
RDM4	100	106	636	0.060
RDM5	100	109	675	0.062
RDM6	100	130	754	0.058
RDM7	98	98	606	0.063
RDM8	400	510	4664	0.022
RDM9	400	660	5926	0.022
RDM10	50	492	4948	0.200
RDM11	50	489	5056	0.200
RDM12	50	492	4954	0.200
RDM13	50	486	4888	0.200
RDM14	50	494	4991	0.200

Problems AIR1 to AIR6 are generated by [POWR 87] and the authors. Problems BUS1 to BUS4 are supplied by Paixao [PAXO 85] and problems BUS5 to BUS8 are supplied by PIM/UNICOM [UNIC 84] and they are of extended set partitioning [DDMT 85] types with side constraints. Problems STR1 and STR2 are taken from [FNHT 74]. Problems RDM1

to RDM14 are supplied by Paixao. They are randomly generated with positive random costs which vary between 1 and 25 for RDM1 to RDM9 and a cost of 1 for problems RDM10 to RDM14. All these models are converted to the standard MPSX format using the modelling system CAMPS [LUMT 87].

3. Preprocessing Heuristics

Preprocessing procedures have been applied by a number of investigators as a preliminary step towards the solution of large scale SP models [BLHO 80] and [GFNH 72]. These procedures may be simple algorithms or heuristics and possess computational complexities which have polynomial time performance. In this section we list a number of procedures which we have adopted to find tight upper and lower bounds for the SCP and to reduce the model size. In this composite approach we start by deriving lower and upper bounds to the SCP using the dual ascent procedure. These bounds are then used as inputs to the lagrangean relaxation. A sufficient number of subgradient iterations are then applied to tighten the bounds and to reduce the model size. At the end of the subgradient optimisation step, the best lagrangean solution is recalled and the row reduction tests are applied. If some rows are removed from the model by the row reduction tests, the subgradient procedure is applied again. At this point if an optimal solution to the SCP is not found we recall the best lagrangean solution already computed and derive a dual feasible solution. This dual solution is then used to derive the costs of the graph relaxation models described in section 4.

The composite procedure is made up of four main procedures [ELDA 88] which are labelled as

(a) dual ascent procedure,
(b) lagrangean and subgradient procedure,
(c) redundant rows procedure
(d) dual solution procedure.

4. Alternative relaxations

A minimising problem Q is said to be a relaxation of a minimising problem P if the set of feasible solutions of Q contains the set of all feasible solutions of P and the corresponding optimal solution value z_Q is less than or equal to the optimal solution value z_p. In proposing alternative relaxations to the SP the main motivation is to derive problems which are easily solved in their relaxed form and provide lower bounds to the SP. Five graph theoretic relaxations [ELDA 88] have been developed for the SP. These can be itemised as

- A network relaxation which can be solved by the greedy method,
- An assignment relaxation based on partitioning the constraint set into two disjoint subsets which represent the set of vertices of the bipartite graph,
- An assignment relaxation using the traveling salesman approach,
- A shortest route relaxation,
- A minimal spanning tree relaxation.

Only the assignment relaxations, the corresponding algorithmic implementation and computational results are considered in this paper.

A lower bound derived by the assignment 1 representation (ASP1)

Let $a_j=(a_1j,...,a_{mj})^T$ denote the column a_j of the SCP problem and let this column be decomposed into a set of k_j columns

$$(a^1_j, ..., a^{k_j}_j)$$

each with at least one unit entry and at most two unit entries. It follows from this decomposition that

$$a_j = \sum_{p=1}^{k_j} a^p_j$$

The column (nonzero) count of a_j^p can only be 1 or 2, that is

$$\sum_{i=1}^{m} a_i^p \in \{1,2\}, p = 1,....,k_j \tag{4.1}$$

where $a_{ij}^p \in \{0,1\}$.

Let q: be the largest positive integer such that $q_j < (|H_j|+1)/2$. The allowable range for k_j is easily seen to be $q_j < k_j < |H_j|$. Let the set R (set of rows) be partitioned into two disjoint sets R' and R" such that R'UR"=R and R'nR" $= \Phi$.

Thus the SCP (1.1-1.3) can be written as

$$\text{Min} \sum_{j=1}^{n} c_j x_j \tag{4.2}$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j > 1 \quad , \qquad i \in R'$$

(4.3)

$$\sum_{j=1}^{n} a_{ij} x_j > 1 \quad , \qquad i \in R" \tag{4.4}$$

$$x_j \in \{0,1\} \quad , \; j = 1,...,n. \tag{4.5}$$

Introduce two (redundant) constraints indexed m+1 and m+2 such that

$$\sum_{j=1}^{n} a_{m+1,\,j} x_j > 0 \tag{4.6}$$

$$\sum_{j=1}^{n} a_{m+2,j} \times j > 0 \tag{4.7}$$

Where $$a_{m+1,j} = \begin{cases} 1 & \text{if } \sum_{i \in R'} a_{ij} | < | \sum_{i \in R^{11}} a_{ij} 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

and $$a_{m+2,j} = \begin{cases} 1 & \text{if } \sum_{i \in R^{11}} a_{ij} | < | \sum_{i \in R'} a_{ij} 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

Thus the SCP can be rewritten as follows

$$\text{Min} \sum_{j=1}^{n} c_j x_j \tag{4.10}$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j > 1, \qquad i \in R' \tag{4.11}$$

$$\sum_{j=1}^{n} a_{ij} x_j > 1, \qquad i \in R^{11} \tag{4.12}$$

$$\sum_{j=1}^{n} a_{m+1,\ j}\ x_j > 0 \tag{4.13}$$

$$\sum_{j=1}^{n} a_{m+2\ ,\ j}\ x_j > 0 \tag{4.14}$$

$$x_j \in \{0,1\} \quad \text{for all} \quad j. \tag{4.15}$$

Let R^{+}=R'U{m+l} and R^{++} = R"U{m+2}. From the definitions set out in (4.1) it follows that it is always possible to derive a decomposition of a_j to
a_j^p (p=l,..,k_j) where $a^p{}_j$ takes one of the three alterative forms.

a)

$$a_j^p = \begin{bmatrix} . \\ a_{rj}^p \\ . \\ o \\ \hline . \\ a_{sj}^p \\ . \\ o \end{bmatrix} \quad \text{where } r \in R' \cap H_j \text{ and } s \in R'' \cap H_j \qquad (4.16)$$

b)

$$a_j^p = \begin{bmatrix} . \\ a_{rj}^p \\ . \\ o \\ \hline o \\ o \\ . \\ a_{m+z,j}^p \end{bmatrix} \quad \text{where } r \in R' \cap H_j \qquad (4.17)$$

c)

$$a_j^p = \begin{bmatrix} o \\ o \\ a_{m+1,j}^p \\ \hline . \\ . \\ a_{sj}^p \\ o \end{bmatrix} \quad \text{where } s \varepsilon R'' \cap H_j \qquad (4.18)$$

In (4.16-4.18) all a_{ij}^p -1 values and the rest of the components are zero.

Assignment relaxation (version 11 of the SCP.

The SCP as presented in (4.10-4.15) can be relaxed as an assignment problem by introducing a bipartite graph with two sets of vertices and arcs as shown below.

Set of vertices. For the m+2 rows introduce m+2 corresponding vertices $v_1,...,v_{m+2}$ which are used in the (assignment) graph representation of the problem.

The set of arcs and associated costs. Introduce three sets of arcs A_B, $A_{D'}$ and $A_{D''}$ defined as

(i) $A_B = \{(v_r, v_s) / r\varepsilon R' \text{ and } s\varepsilon R''\}$

There is an arc from v_r to v_s if there exists a column a_j^p which satisfies (4.16), with the associated cost $d_{jp} - 2c_j / |H_j|$.

(ii) $A_{D''} - \{(v_r, v_{m+2}) \setminus r\varepsilon R'\}$.

There is an arc from v_r to v_{m+2} if there exists a column a_j^p which satisfies (4.17) with the associated cost $d_{jp} = c_j / |H_j|$.

(iii) $A_{D'} = \{(v_{m+1}, v_s) | s \in R''\}$.

There is an are from v_{m+1} to v_s if there exists a column a_j^p which satisfies (4.18). The associated cost d_{jp} is given as $d_{jp} = c_j / |H_j|$

Finally introduce an arc set A_{n+1} with a dummy arc from v_{m+1} to v_{m+2}, $A_{n+1} = \{(v_{m+1}, v_{m+2})\}$ with zero associated cost $d_{n+1,1} = 0$, whereby the flow requirements (>1) may be imposed on v_{m+1} and v_{m+2}. Let a_{n+1} denote the corresponding dummy column. Let A denote the directed arcs of the resulting graph such that $A=\{A_B \cup A_{D'} \cup A_{D''} \cup A_{n+1}\}$. Figure 4.1 illustrates the structure of this graph. Let A_j denote the set of arcs obtained by decomposing the column j in the manner indicated earlier. Thus there are k_j arcs in A_j, where for $(v_r,v_s) \in A_j$, $v_r \in R^+$ and $v_r \in R^{++}$ as explained in (4.16-4.18). It is easy to see that

$$\bigcup_{j=1}^{n+1} A_j = A .$$

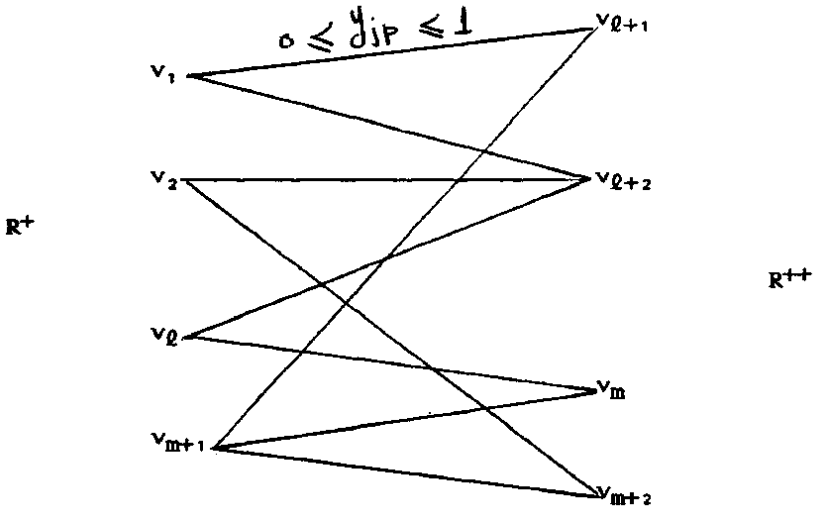


Figure 4.1

A statement of the relaxed problem [FRZE 85]. With each column a_j associate a variable $y_{jp}, y_{jp} \in \{0,1\}$ and a cost coefficient d_{jp} which are defined for j=l,...,n+l and $p = 1,....,k_j$. Then the relaxed problem ASP1 is stated as

$$\text{Min} \quad \sum_{j=1}^{n+1} \sum_{p=1}^{k_j} d_{jp} y_{jp} \tag{4.19}$$

subject to

$$\sum_{j=1}^{n+1} \sum_{p=1}^{k_j} \alpha^p_{r s_j} y_{jp} > 1, \; r \in R^+ \tag{4.20}$$

$$\sum_{j=1}^{n+1} \sum_{p=1}^{k_j} \alpha^p_{r s_j} y_{jp} > 1, \; s \varepsilon R^+ \tag{4.21}$$

$$y_{jp} \in \{0,1\} \quad \text{for all j and p} \tag{4.22}$$

where $$\alpha^{p}_{rsj} = \left\{ \begin{matrix} 1 \text{ if } (v_r, v_s) \in A_j \\ 0 \text{ otherwise} \end{matrix} \right\} \; j=1,\ldots,n+1 \tag{4.23}$$

ASPl set out in (4.19-4.23) is a proper relaxation of the SCP and is explained below.

Let the cost coefficient of ASPl, dip be defined such that

$$\sum_{p=1}^{k_j} d_{jp} = c_j . \tag{4.24}$$

(a) For any x* which is a feasible solution to the SCP it is simple to construct a feasible solution y* to the ASPl in the following way

$$\left. \begin{matrix} \text{for } x^*_j = 0 & \text{set} & y_{j1} = \ldots = y^*_{jk_j} = 0 \\ \text{and} & & \\ \text{for } x^*_j = 1 & \text{set} & y_{j1} = \ldots = y^*_{jk_j} = 1 \end{matrix} \right\} \; j=1,\ldots,n$$

also set $y_{n+1,1} = 1$. This implies that the following relation for those columns a_j which are decomposed such that $k_j > 2, j = 1,\ldots, n$

$$y_{jp} = y_{jp+1}, \; p = 1,\ldots,k_j - 1 \tag{4.25}$$

must also be satisfied. The solution x* for the SCP and y* for the ASPl have the same objective function value (see 4.24).

(b) If the optimal solution $\bar{y}$ to the ASPl also satisfies (4.25) then this implies that the vector $\bar{x}$ computed by the relations

$$\bar{x}_j = \bar{y}_{jp} \qquad j=1,..,n, \qquad p=1,\ldots,k_j \tag{4.26}$$

is also an optimum solution to the SCP. If the optimal solution to the ASPl does not satisfy (4.25) it still provides a valid lower bound on the optimal objective value of the SCP. At this stage a tree search method can be used to satisfy the relations (4.25) by suitably fixing groups of y_{jp} to zero or one.

Let $\lambda_j = (\lambda_{j1},...,\lambda_{jk_j})$, $j=1,\ldots,n$ represent the lagrangean multipliers associated with the side constraints of the decomposed column a_j (j=l,...,n) where λ_j are unrestricted in sign. Let $\lambda = (\lambda_1,...,\lambda_n)$ represent a collection of lagrangean multipliers The lagrangean relaxation [GOEF 74] LASPl(λ) of the ASPl as set out in (4.19-4.23) and (4.25) can be written as

$$\text{Min} \sum_{j=1}^{n} \sum_{k=1}^{k_j} y_{jk}(d_{jk} + \lambda_{jk} - \lambda_{jk-1})$$

subject to

network constraints (4.20-4.21)

and

$$y_{jk} \in \{0,1\}, \quad j=1,...,n \text{ and } k=1,...,k_j$$

where $y_{jo} = \lambda_{jk_j} = 0, \quad j=1,...,n$

If we replace the inequality (">") in (4.20) and (4.21) by equality ("=") for $r \in R'$ and $s \in R''$ respectively then the modified ASPl becomes a relaxation to the SPP. This implies that the constraints corresponding to $i \varepsilon R$ must be strictly equal to 1, while the two constraints corresponding to $i \varepsilon$ {m+l,m+2} can be > 1.

A lower bound derived by the assignment 2 representation (ASP2)

Let kj = $|H_j|$ denote the number of 1's in the column a_j Let H_j as defined in (1.4) be re-expressed as $H_j = \{i_1,...,i_{k_j}\}$. Introduce the vertex set V corresponding to the rows i=l,...,m of the SCP such that V={v,,...,v_m}. For each column a_j construct a set of k_j directed arcs from V to V which admit unit flow in the following way. The arc sets in the equivelent bipartite graph are denoted by A_j where

$$A_j = \left\{(v_{i_1}, v_{i_2}),(v_{i_2}, v_{i_3}),...,(v_{i_{k_j-1}}, v_{i_{k_j}}), (v_{i_{k_j}}, v_{i_1})\right\} \ , \ j = 1 \ , \ldots , n \qquad (4.27)$$

Let the associated cost for each arc in the arc set A_j be defined as

$$d_j = c_j / k_j. \qquad (4.28)$$

With each of the k_j arcs taken from the set A_j associate a variable

$y^j_{pq} \in \{0,1\}$ and defined for all $(v_p,v_q) \in A_j$. (4.29)

Let $$\alpha^j_{pq} = \left\{\begin{matrix} 1 \text{ if } (v_p,v_q) \ \varepsilon \ A_j \\ 0 \text{ otherwise} \end{matrix}\right\} \ j=1,\ldots,n \qquad (4.30)$$

The relaxed problem ASP2 is stated as follows.

$$\text{Min} \sum_{j=1}^{n} \sum_{(v_p,v_q)\varepsilon A_j} d_j y^j_{pq} \qquad (4.31)$$

subject to

$$\sum_{j=1}^{n} \alpha^j_{iq} \ y^j_{iq} \ > \ 1 \ , \ i = 1,...,m \qquad (4.32)$$

$$\sum_{j=1}^{n} \alpha^j_{pi} \ y^j_{pi} \ > \ 1 \ , \ i = 1,...,m \qquad (4.33)$$

$$y^{j}_{pq} \varepsilon \{0,1\}, \quad j = 1,...,n \text{ and } (v_p, v_q) \varepsilon A_j \tag{4.34}$$

ASP2 set out in (4.31-4.44) is a proper relaxation of the SCP. This is explained below.

(a) for any x* which is a feasible solution to the SCP it is simple to construct a feasible solution y* to the ASP2 by the procedure

$$\left.\begin{array}{llll} \text{for} & x^{*}_{j} = 0 & \text{set} & y^{*j}_{pq} = 0, \ (v_p, v_q) \varepsilon A_j \\ \text{and} & & & \\ \text{for} & x^{*}_{j} = 1 & \text{set} & y^{*j}_{pq} = 1, \ (v_p, v_q) \varepsilon A_j \end{array}\right\} \ j=1,\ldots,n$$

This implies that the following relations for those columns a_j such that $|H_j| > 2$, j=l,...,n

$$y^{j}_{i_1 \ i_2} = y^{j}_{i_2 \ i_3} = ... = y^{j}_{i_{k_j} \ i_1} \tag{4.35}$$

must also be satisfied. The feasible solution y* to the ASP2 has a cost which is equal to the cost of the feasible solution x* of the SCP (see 4.28).

(b) If the optimal solution $\bar{y}$ to the ASP2 also satisfies (4.35) then this implies that the vector $\bar{x}$ computed by the relations

$$\bar{x}_j = \bar{y}^{j}_{pq}, \quad j = 1,...,n \text{ and } (v_p, v_q) \varepsilon A_j \tag{4.36}$$

is also an optimum solution to the SCP. If the optimal solution to the ASP2 does not satisfy (4.35), it still provides a valid lower bound on the optimal objective value of the SCP. At this stage a tree search method can be used to satisfy the relations (4.35) by

suitably fixing groups of k_j variables as defined in (4.35) to the values zero or one. Let $\lambda_j = (\lambda_{j1},...,\lambda_{jk_j})$, $j = 1,\ldots,n$ represent the lagrangean multipliers associated with the side constraints of the decomposed column a_j (j=l,...,n) where λ_j are unrestricted in sign. Let $\lambda = (\lambda_1,...,\lambda_n)$ represent a collection of lagrangean multipliers The lagrangean relaxation [GOEF 74] LASP2(λ) of the ASP2 as set out in (4.31-4.34) and (4.35) can be written as

$$\text{Min} \quad \sum_{j=1}^{n} \sum_{k=1}^{k_j} y^j_{i_k i_{k+1}} (d^j_{i_k i_{k+1}} + \lambda_{jk} - \lambda_{jk-1})$$

subject to

network constraints (4.32-4.33)

and

$$y^j_{pq} \varepsilon \{0,1\}, \quad j = 1,...,n \text{ and } (v_p, v_q) \varepsilon A_j$$

where $\lambda_{jo} = \lambda_{jk_j} = 0$, $i_{k_{j+1}} = i_1$, $j = 1,...,n$.

If we replace the inequality ">" in (4.32-4.33) by equality "="H then the modified ASP2 becomes a relaxation to the SPP.

5. A unified tree search algorithm

Branch and bound is one of the most successful techniques for solving combinatorial optimisation problems, covering discrete optimisation models in general and integer programming problems in particular [MRTY 76]. The computational efficiency of the tree developement and the search procedure which follows from branch and bound depends on a number of factors which are considered below.

In designing the branch and bound algorithm, we wish to control the size of the tree

developed and the process of searching for the optimal solution of the original problem, that is, controlling the number of subproblems proposed. This search strategy depends on the heuristics as defined by the choices made at the following algorithmic steps.

(a) The type of the relaxation used to represent the original problem (problem representation). This also influences the time taken to solve each subproblem (reoptimise).
(b) The choice of the branching variable (branching strategy).
(c) The choice of the best subproblem to solve (search strategy).

In this context Breu et al [BRBD 74] have surveyed the art and science of branch and bound techniques for 0-1 integer programming. Shapiro [SHAP 79] has incorporated the lagrangean relaxation within the framework of the tree search procedure and Mitra [MTRA 73] has investigated different strategies for the tree search procedure as applied to mixed integer programming.

To start with one of the relaxed SCP problems, namely, ASP1 or ASP2 is solved at the root node. After solving the subproblem the solution is analysed and a variable is chosen for branching. The choice of a branching variable is a difficult task and no universal rule exists which gives a uniformly good result. Many investigators have remarked that, this choice rule should be formulated by studying the problem and its structure. By and large good branching rules are highly context dependent. In choosing the next subproblem from the waiting list of subproblems held in a stack, again many alternative criteria can be used and together with the variable choice strategies, these determine the size of the search tree. In our investigation we have taken the most popular method of last in first out (UFO) rule.

An outline of the unified tree search strategy

The alternative strategies which are adopted in the design of the branch and bound procedure are outlined in the statement of the algorithm set out below. At each step of the solution process it is necessary to know whether or not a feasible solution to the SCP is obtained. It is also necessary to know the value of the best objective function which is denoted by z_{min}. Also we use $z\ell$ to denote the optimal solution value of the relaxed problem (ASP1 or ASP2), that is, a lower bound on the objective function value of the SCP. Before stating the algorithm we explain how the search is terminated at each branch of the tree depending on the outcome of the subproblem investigation. A node is fathomed if after the solution of the subproblem one of the following conditions hold.

(1) The lower bound is greater than the upper bound.

(2) The subproblem has no feasible solution.

(3) The optimal solution to the relaxed problem is found and this also satisfies the side contraints. Hence it is a feasible solution to the SCP and because of (1) above it is also the best feasible solution to the SCP found so far.

Step(l) Preprocessing procedure

In this step the composite preprocessing procedures (a), (b) and (c) outlined in section 3 are applied to reduce the model size and to derive an upper bound (z_{min}) and a lower bound (z_ℓ) on $\bar{z}$. If $z_\ell > z_{min} - 1$ go to Step(11), otherwise apply the preprocessing procedure (d) to obtain a dual feasible solution, which is used to derive the cost vector of the relaxed problem.

Step(2) Solution at the root node

The relaxed SCP problem is solved at the root node of the tree. If $z_\ell > z_{min} - 1$ or the side constraints are satisfied go to Step(11), (the optimal solution to the SCP is obtained which is equal to z_{min} in the former and z_ℓ in the latter), else initialise the lagrangean multipliers and go to Step(8).

Step(3) Choice of the branching variable(s)

Select a group of network variables y_{jk} corresponding to the arcs in the arc set

A_j for branching. Add the two subproblems with $(y_{jk} = 1, k = 1,..., k_j)$ and $(y_{jk} = 1, k = 1,..., k_j)$ to the list of subproblems, store their positions in the list and go to Step(4b).

Step(4) Subproblem selection

(a) If the list of subproblems is empty go to Step(11).

(b) Choose a subproblem from the list.

Step(5) Subproblem preparation

Prepare the subproblem to be optimised, that is, identify the out-of- kilter arcs, for example, update the list of subproblems, etc.

Step(6) Subproblem solution

Solve the subproblem using a network optimiser [FDFL 62].

Step(7) Subproblem solution analysis

If the subproblem has no feasible solution or the objective solution value is greater than Z_{min} -1 then go to Step(4). If the side constraints are satisfied go to Step(10). Otherwise go to Step(8).

Step(8) Solution improvement and model reduction

Test for optimal network solution improvement. Temporary remove the redundant columns using reduced cost analysis and single row tests.

Step(9) Lagrangean and subgradient procedure

If the number of subgradient iterations exceed an iteration counter (LMAX) go to Step(3). Otherwise compute the lagrangean multipliers, update the corresponding costs, set the negative costs to nonnegative costs and go to step(5).

Step(10) Update the best SCP solution

A feasible solution to the SCP has been found. Update z_{min} and the corresponding solution vector and go to Step(4).

Step(11) SCP optimal solution

Output the best SCP feasible solution and the corresponding optimal objective value (z_{min}).

6. Computational results

The computational results to the branch and bound algorithm designed for the ASP1 and ASP2 are given in tables 6.1 and 6.2 respectively. This algorithm is applied to the test problems which have not been optimaly solved using the preprocessing procedures of section 3. The abbreviations used in these tables headings are explained below.

$\bar{z}$ is the optimal solution value if found. A value with a "*" represents the best feasible solution found within the time limit.

m, n are the reduced dimensions of the test problems, where m denotes the number of rows and n denotes the number of columns.

z_u is the best upper bound derived using the preprocessing procedures. The computing time taken of the preprocessing procedures is given in the next column. The bound derived by the ASP1 and ASP2 at the root node of the tree and the corresponding computing time are also reported in these tables, followed by the total number of nodes developed in the process of searching for the optimal solution. The number of lagrangean iterations applied within the framwork of the tree search is also tabulated. The last two columns in these tables represent the total time for the branch and bound algorithm (not including step(l)) and the total execution time. The maximum running time permitted was set to 1000 cpu seconds.

Computational results of processing the test problems by the tree search method on ASP1.

PROBLEM		REDUCED DIMENSIONS			PREPRO-	INITIAL			NO OF		TOTAL
NAME	Z	m	n	Z_u	CESSING TIME	NETWORK SOLUTION	TIME	NO OF NODES	LAG ITER	B&B TIME	TIME
AIR1	16635	120	297	16660	40.93	16577.2	6.12	1136	601	954.1	995.0
AIR2	18880	129	436	18885	269.3	18833.1	7.3	26	11	67.0	336.3
AIR3	18195	104	499	18195	175.4	18124.0	6.9	145	113	528.3	703.7
AIR4	19715	138	1089	19795	100.9	19462.7	16.57	225	144	197.9	298.8
AIR5	21560	131	1003	21800	104.55	21377.3	14.67	357	173	342.7	347.2
AIR6	16925	124	998	17000	124.3	16795.0	14.55	31	0	44.1	168.4
BUS2	41051	26	90	41537	1.52	41036.4	.04	16	2	.87	2.39
BUS3	64749*	55	977	64749	62.31	63000.2	7.6	487	539	>1000	>1000
BUS4	74787*	53	547	75212	26.63	72086.1	2.6	818	966	>1000	>1000
RIM4	97	99	74	97	11.5	93.23	.21	75	30	45.4	56.9
RIM5	96	31	54	96	10.7	94.37	.12	37	30	34.8	45.5
RIM6	87	57	40	87	15.2	85.6	.59	70	31	16.3	31.5

Table 6.1

All times are in seconds of Honeywell Multics DP6840 processing

Computation results of processing the test problems by the tree search method based on ASP2.

PROBLEM NAME	REDUCED DIMENSIONS z	m	n	z_n	PREPRO-CESSING TIME	INITIAL NETWORK SOLUTION	TIME	NO OF NODIES	NO OF LAG ITER	B&B TIME	TOTAL TIME
AIR1	16635	120	297	16660	40.93	16577.5	14.43	454	172	>1000	>1000
AIR2	18880	129	436	18885	269.3	18835.2	24.2	110	74	893.3	1162.6
AIR3	18195	104	499	18195	175.4	18136.5	18.6	88	11	283.3	458.66
AIR4	19715	138	1089	19795	100.9	19469.8	12.51	177	135	>1000	>1000
AIR5	21560	131	1003	21800	104.55	21379.0	11.5	403	231	>1000	>1000
AIR6	16925	124	998	17000	124.3	16899.0	25.4	49	13	577	701.3
BUS2	41051	26	90	41537	1.52	41046.4	.1	22	7	1.7	3.22
BUS3	64278*	55	977	64749	62.31	63029.0	39.6	116	122	>1000	>1000
BUS4	73694*	53	547	75212	26.63	72125.2	9.9	142	210	>1000	>1000
RIM4	97	99	74	97	11.5	93.23	.89	75	94	584.0	600.0
RIM5	96	31	54	96	10.7	94.37	.87	46	14	94.5	115.2
RIM6	87	57	40	87	15.2	85.7	.67	47	7	34.1	49.3

Table 6.2
All times are in seconds of Honeywell
Multics DP6840 processing

The ASP1 solved nearly all the test problems within the time limit, except for BUS3 and BUS4. The ASP2 fails to solve problems AIR1, AIR4, AIRS, BUS3 and BUS4 within this time limit. The lagrangean and subgradient procedures manage to improve the bounds at the lower levels of the tree, that is, when a large number of variables have been fixed to either one or zero, but, at the expense very high computing time. The best feasible solutions obtained for BUS3 and BUS4 test problems by the ASP2 are better than these obtained by the ASP1.

Overall more experiments are needed to "successfully" incorporate the lagrangean and subgradient procedures within the tree search procedure. Different branching strategies also may be worth investigating and of course a faster network optimiser will improve the performance of this algorithm.

REFERENCES

[BAFS 81] Baker, E., and Fisher, M., "Computational results for very large air crew scheduling problems", Omega, 9, pp613-618, (1981)

[BALS 83] Balas, E., "A class of location, distribution and scheduling problems: modelling and solution methods". Revue Beige de Statistique, d'Informatique et de Recherche Operationnelle, 22, pp36-57, (1983)

[BLHO 80] Balas, E., and Ho, A., "Set covering algorithms using cutting planes, heuristics and subgradient optimization: A new computational study", Mathematical Programming Study, 12, pp37-60, (1980)

[BLPD 76] Balas, E., and Padberg, M.W., "Set partioning: A survey", SIAM Review, 18, pp710-760, (1976)

[BRBD 74] Breu, R., and Burdet, C.A., "Branch and bound experiments in 0-1 programming", Mathematical Programming Study, 2, ppl-50, (1974)

[CHRS 85] Christofides, N., "Vehicle routing" in "Travelling salesman problem", Lawler, E.L., Lenstra, J.K., Rinooy Khan, A.H.G., and Shmays, D.S., eds, Academic Press, (1985)

[DANT 63] Dantzig, G., "Linear programming and extensions", Princeton University Press, Princeton, New Jersey, (1963)

[DDMT 85] Darby-Dowman, K., and Mitra, G., "An extension to set partitioning with application to scheduling problems", European Journal of Operations Research, 21, pp200-205, (1985)

[DKST 81] Dasltin, M.S., and Stern, E.D., "A hierarchical objective set covering model for emergency medical service vehicle deployment" Transportation Science, 15, ppl37-152, (1981)

[EDMT 88] E, El-Darzi., and G, Mitra., "Set covering and set partitioning: A collection of test problems", Brunei University, Internal Report, (1988)

[ELDA 88] E, El-Darzi., "Methods for solving the set covering and set partitioning problems using graph theoretic (relaxation) algorithms", Brunei University, PhD thesis, (1988)

[FDFL 62] Ford, L., and Fulkerson, D., "flows in networks", Princeton University Press, (1962)

[FNHT 74] Fulkerson, D.R., Nemhauser, G.L., and Trotter, L.E., "Two computationally difficult set covering problems that arises in computing the 1-Width of incidence matrices of steiner triple systems", Mathematical programming Study, 2, pp72-81, (1974)

[FRZE 85] Frieze, A., "Private communication", (1985)

[GEOF 74] Geoffrion, A.M., "Lagrangean relaxation for integer programming", Mathematical Programming Study, 2, pp82-114, (1974)

[GFNH 72] Garfinkel, R.S., and Nemhauser, G.L., "Integer programming", Wiley, (1972)

[LUMT 87] Lucas, C, and Mitra, G., "Computer assisted mathematical programming modelling system (CAMPS)" User Specification, Brunei University, Department of Mathematics and Statistics, (1987)

[MRTY 76] Murty, K.G., "Linear and combinatorial programming", John Wiley and Sons, (1976)

[MTDD 85] Mitra, G., and Darby-Dowman, K., "CRU-SCHED A computer based bus crew scheduling system using integer programming", in "Computer scheduling of public transport", Rousseau, J.-M., Ed., North Holland, (1985)

[MTRA 73] Mitra, G., "Investigation of some branch and bound strategies for the solution of mixed integer linear programs", Mathematical Programming, 4, ppl50-170, (1973)

[PAXO 85] Paixao, J., "Private communication", (1985)

[POWR 87] Powers, D., "Investigation and construction of set covering and set partitioning test problems", BSc dissertation, Brunei University, (1987)

[SHAP 79] Shapiro, J.F., "A survey of lagrangean techniques for discrete optimization", Annals of Discrete Mathematics, 5, pp113-138, (1979)

[UNIC 84] "Private communication", Unicom Consultancy, Brunei Science Park, (1984)

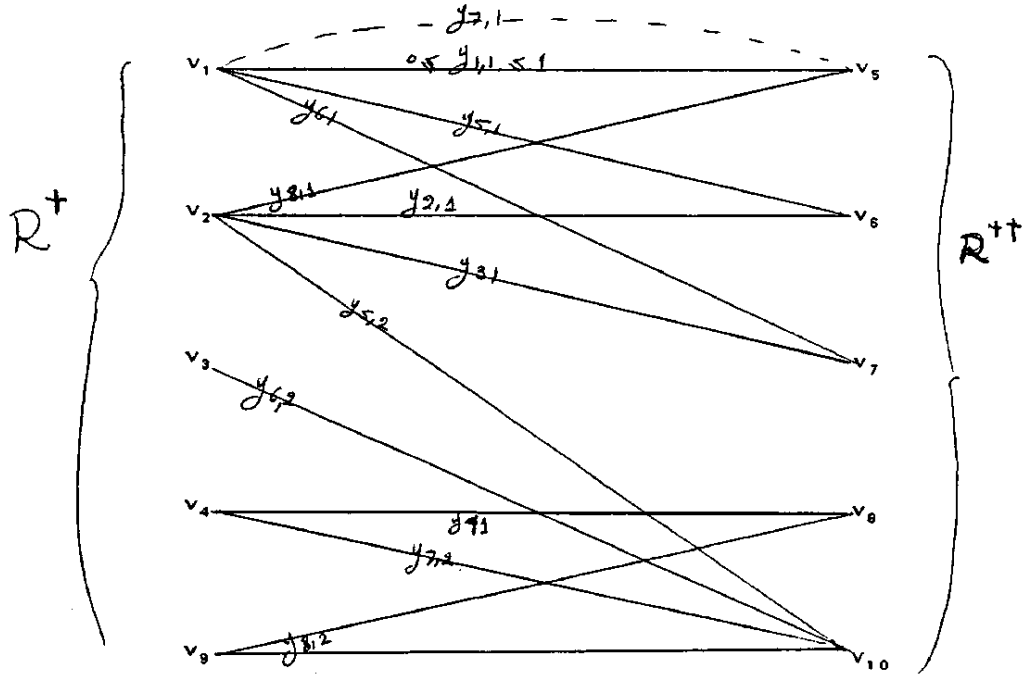
[WILL 85] Williams, H.P., "Model building in mathematical programming", John Wiley and Sons, (1985)

Appendix

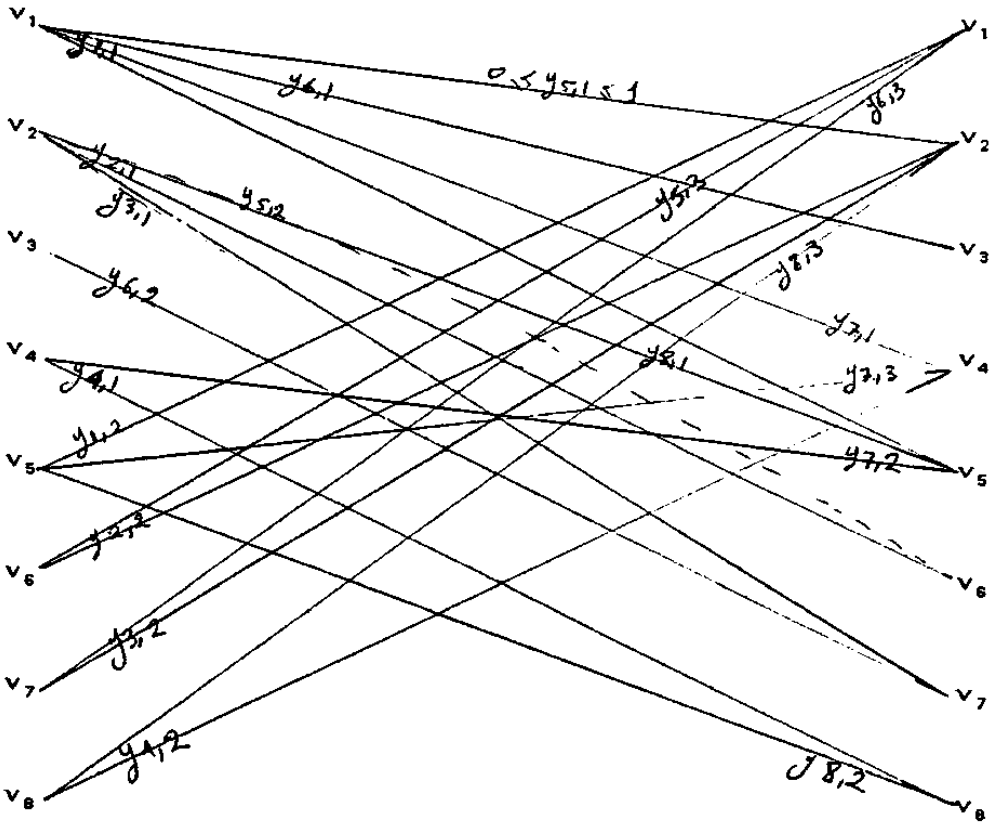
ILLUSTRATIVE EXAMPLE OF THE ALTERNATIVE RELAXATIONS

In this appendix the alternative relaxations of the SP (ASPl, ASP2) are illustrated .using the SCP example set out in the tableau below. variables. y_{jk},k=l,...k_j reprsent the variables for the assignment relaxations, where k_j denotes the total number of arcs derived from column a: of the SCP.

	c 0 1 1	c 0 1 2	c 0 1 3	c 0 1 4	c 0 1 5	c 0 1 6	c 0 1 7	c 0 1 8		
cost	4	3	3	2	3	2	3	4		
row1	1				1	1	1		>	1
row2		1	1		1			1	>	1
row3						1			>	1
row4				1			1		>	1
row5	1						1	1	>	1
row6		1			1				>	1
row7			1			1			>	1
row8				1				1	>	1



The ASP1 representation of the SCP example



The ASP2 representation of the SCP example