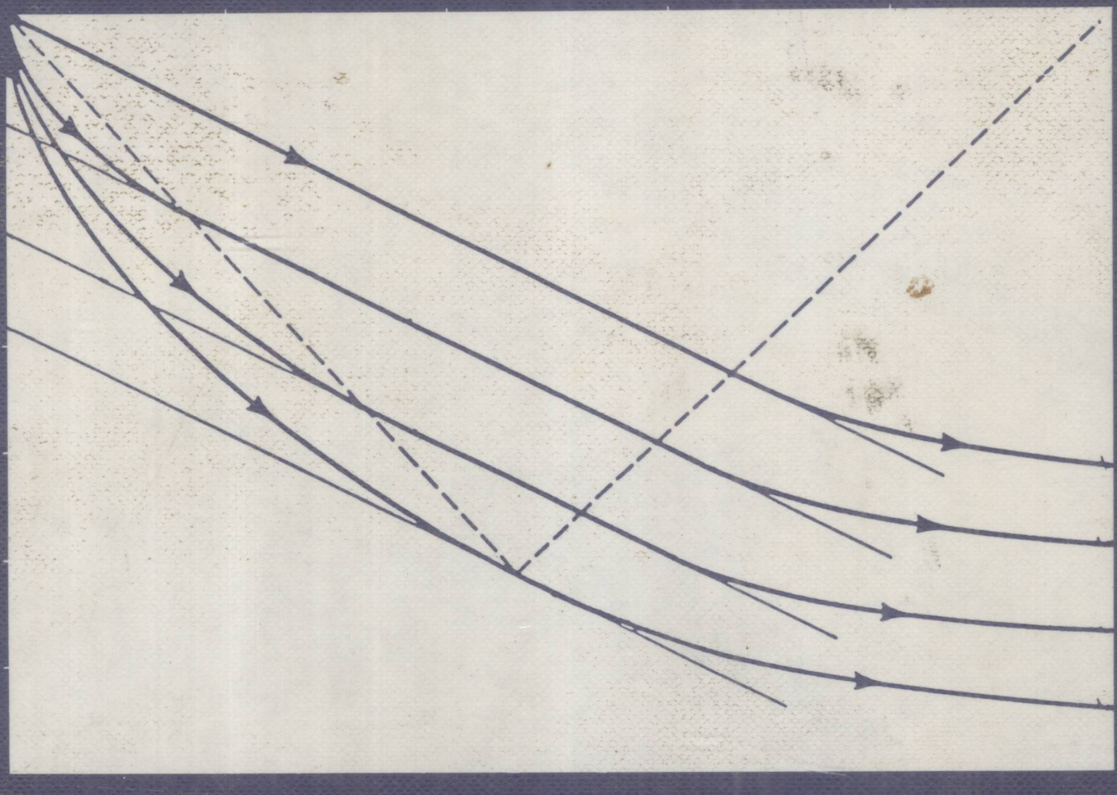
OPTIMIZATION
TECHNIQUES AND APPLICATIONS



Edited by
K. L. TEO
H. PAUL
K. L. CHEW
C. M. WANG

The Proceedings of International Conference on Optimization: Techniques and Applications

ICOTA

8 — 10 APRIL, 1987

Edited by

K. L. Teo **H. Paul**
Department of Industrial and Systems Engineering
National University of Singapore
Singapore

K. L. Chew
Department of Mathematics
National University of Singapore
Singapore

C. M. Wang
Department of Civil Engineering
National University of Singapore
Singapore

The ICOTA Organising Committee is not responsible for the statements made or for the opinions expressed in the papers included in this proceedings.

ISBN 9971-62-129-0

Preface

Optimization is concerned with the most efficient allocation of scarce resources among competing needs for them, and generally leads to a best plan, design, or operational procedure. Optimization techniques are employed in solving a broad range of management, engineering, industrial, military, government, and socio-economic problems. Some of the commonly used optimization techniques are linear programming, integer programming, nonlinear programming, dynamic programming, network optimization methods, and optimal control.

A large number of application software packages are now available for solving various real-world optimization problems encountered in industry, business and government projects. For instance, the petroleum industry makes extensive use of linear programming techniques in scheduling, distribution and product blending whilst integer programming is being increasingly applied to complex problems of scheduling operations. Dynamic programming has proved its worth in solving complex replacement and maintenance problems, and optimal control has proven applications in solving many management and engineering problems such as optimal advertising policy, minimizing production cost, maximizing profit, or moving the arm of a robot from one point to another point in minimum time.

With a view to bringing together researchers and practitioners and to provide a platform for the interchange of information on the multidisciplinary approaches to optimization, ten academic departments/schools of the National University of Singapore have jointly organised this International Conference on Optimization Techniques and Applications (ICOTA). It covers all aspects of optimization, and it is the first such conference in the South East Asian region.

There were some 160 participants from 28 countries. 114 papers were presented at the Conference. This book contains all the presented papers. This proceedings is divided into the following sections:

- Invited Lectures
- General Lectures
- Computer Science
- Communication Engineering
- Construction Engineering
- Control and Systems
- Engineering Design
- Electrical and Power Systems
- Graphs and Networks
- Industrial Engineering
- Integer Programming
- Manufacturing Systems
- Multicriteria Optimization
- Management
- Nonlinear Mathematical Programming
- Optimal Control
- Structural Optimization
- Transportation and Production Scheduling
- Water Resources

It is hoped that the papers will be of interest to researchers and practitioners in the area of optimization and that these will provide some useful information on recent developments in the area.

In editing these proceedings, we have been assisted by the ICOTA Committee members. Financial support from the National University of Singapore, Singapore Turf Club and Lee Foundation Singapore, in organizing the Conference are gratefully acknowledged. We would also like to thank all those who have in one way or another contributed towards the success of the Conference. Finally, special thanks are due to Professor H. H. Huang, Deputy Vice-Chancellor, National University of Singapore for his help in making the Conference a success.

The Editors

TABLE OF CONTENTS

Electrical Machines and Power Systems

Engineering Design

Graphs and Networks

Industrial Engineering

Integer Programming

Management

Manufacturing Systems

Multicriteria Optimization

Nonlinear Mathematical Programming

Optimal Control

Structural Optimization

Transportation and Production Scheduling

Water Resources

Appendix

Invited Lectures

International Conference on Optimization Techniques and Applications
Singapore, 8 — 10 April 1987, pp. 2 — 35

ALGORITHMS FOR LARGE SCALE SET COVERING PROBLEMS

Nicos Christofides

Department of Management Science
Imperial College of Science and Technology
Exhibition Road, London SW7 2BX
United Kingdom

ABSTRACT. This paper is concerned with the set covering problem (SCP), and in particular with the development of a new algorithm capable of solving large-scale SCP's of the size found in real life situations.

The set covering problem has a wide variety of practical applications such as crew scheduling, vehicle dispatching, facility location, information retrieval, political districting, design of switching circuits and others. A common feature of most of these applications is that they yield large and sparse SCPs normally with hundreds of rows and thousands of columns. In this paper we present an algorithm capable of solving problems of this size and test problems up to 400 rows and 4000 columns are solved and results reported. This is by far the largest size SCP reported solved in the literature.

The method developed in this paper consists of a combination of decomposition and state space relaxation which is a technique recently developed for obtaining lower bounds on the dynamic program associated with a combinatorial optimization problem. The large size SCP's are decomposed into many smaller SCP's which are then solved or bounded by state space relaxation (SSR). Before using the decomposition and SSR, reductions both in the number of columns and the number of rows of the problem are made by applying lagrangean relaxation, linear programming and heuristic methods.

1. INTRODUCTION

1.1 Manuscripts

The set covering problem is a well-known combinatorial optimization problem consisting of finding a subset of columns of a zero-one mxn matrix such that it covers the rows of the matrix, at a minimum cost. The problem is formally stated as the integer program:

$$\text{(SCP)} \quad \min \sum_{j=1}^{n} c_j x_j$$

$$\text{st.} \quad \sum_{j=1}^{n} a_{ij} x_j \geq 1 \qquad (i=1,\ldots,m)$$

$$x_j \in \{0,1\} \qquad (j=1,\ldots,n)$$

where $c_j>0$ is the cost of column j and $A=[a_{ij}]$ is the mxn zero-one matrix.

Let a_i and a^j denote respectively the i-th row and the j-th column of A, and let $M=\{1,2,\ldots,m\}$ and $N=\{1,2,\ldots,n\}$. We denote $M_j= \{i \in M : a_{ij}=1\}$ and $N_i=\{j \in N : a_{ij}=1\}$.

The SCP and related problems, have a wide field of applications such as crew scheduling ([1], [2], [9], [27], [32], [35], [37]) vehicle dispatching ([11]), facility location ([8], [10], [14], [26], [36], [39]), information retrieval ([17]), political and health districting ([16], [21], [23]), assembly line balancing ([38]), minimisation of boolean expressions ([24], [40]), and others ([7]). See [12] for a computational study of the SCP and [20] and [5] for detailed surveys.

The SCP also has particularly strong links with other combinatorial optimization problems, especially graph theoretic problems ([34]).

Several algorithms have been proposed for the SCP and can be found in [3], [4], [6], [12], [19], [30]. In this paper, we present and apply an algorithm for larger size SCP's consisting of a combination of some of the methods used before for solving the problem, with decomposition and state space relaxation - which is a technique recently developed for obtaining lower bounds on dynamic programs ([13], [34]).

In section 2, a procedure to perform reductions on the dimension of the problem is presented and computationally tested for large size SCP's. The method consists of a combined utilisation of lagrangean relaxation, linear programming and heuristics. In section 3, state space relaxation is applied to the computation of lower bounds for the SCP and these values are compared with the LP bounds for different types of test problems. In section 4, we develop a decomposition technique for large size SCP's which is combined with state space relaxation obtaining bounds on the optimal value to the SCP. Computational results are also shown for large scale problems and for unicost (all $c_j=1$) SCP's. Finally, in section 5, we present a tree-search algorithm making use of the techniques described in the previous sections and full computational results for problems with up to 400 rows and 4000 columns are given.

2. REDUCTIONS FOR LARGE SIZE SET COVERING PROBLEMS

In this section, a procedure to perform reductions on the number of columns and rows of large size SCP's is presented, and computational results are shown for three different classes of test problems.

2.1 Preliminary Reductions

Before using an algorithm for solving the SCP a number of preliminary reductions can be made both on the columns and rows of the problem. These are well known in the literature and include simple row and column dominance tests [20].

Some of these reductions are unlikely to be useful for a practical SCP, but if the problem occurs as a sub-problem of some other SCP, namely in a tree-search procedure or using any sort of relaxation, then they can be very effective. The row dominance test may also be useful for randomly generated problems in which the coefficients a_{ij} are independent random variables with a fixed probability that $a_{ij}=1$. The column dominance although useful for several practical examples implies an expensive computational effort in terms of time for large scale SCP's. Note that most of the preliminary reductions mentioned in [20] are not useful for unicost SCP's or when c_j is proportional to the number of 1's in column j.

2.2 **Linear Programming Relaxation**

The linear programming relaxation of the SCP is:

$$\text{(LP)} \qquad \min \ \Sigma_{j \varepsilon N} \ c_j \ x_j$$

$$\text{st.} \ \Sigma_{j \varepsilon N_i} \ x_j \geq 1 \qquad i \varepsilon M)$$

$$x_j \geq 0 \qquad (j \varepsilon N)$$

If v(LP) and v(SCP) are the optimum values of LP and SCP respectively then v(LP)≤v(SCP) and if the LP solution is integer then it is the optimal solution to the SCP.

The dual linear relaxation of the SCP is then:

$$\text{(DLP)} \qquad \max \ \Sigma_{i \varepsilon M} \ u_i$$

$$\text{st.} \ \Sigma_{i \varepsilon M_j} \ u_i \leq c_j \qquad (j \varepsilon N)$$

$$u_i \geq 0 \qquad (i \varepsilon M)$$

If DLP is the dual problem to LP and u_i are the dual variables then reduced costs for the variables x_j are given by:

$$s_j = c_j - \Sigma_{i \varepsilon M_j} \ u_i \qquad (2.1)$$

and the variable x_j can be removed from the SCP if

$$s_j + \Sigma_{i \varepsilon M} \ u_i \geq z_u \qquad (2.2)$$

with z_u a known upper bound on v(SCP).

2.3 **Heuristics**

Heuristic algorithms can be used to obtain both upper and lower bounds on v(SCP). Tight upper bounds are important in removing variables from reduced cost analysis and also in fathoming tests when a tree-search procedure is being used to solve the problem. Greedy heuristics to obtain upper bounds to the SCP have been studied in detail in [4] whose main conclusions were adopted in the algorithm used in this paper. Although these greedy-type heuristics have a theoretically poor worst case performance ([15], [28]) they have proved reasonably good for many test problems. If reduced costs s_j for the variables are available, then the greedy heuristics can be applied on the basis of s_j instead of c_j. In [4] this technique is reported as having consistently improved the original "greedy" upper bounds. As will be seen later in this section, our computational experience confirms that result.

Later on in this paper, we present a heuristic based on the decomposition of large size SCP's, which yield, in many cases, an improvement on the upper bounds mentioned above.

2.4 **Lagrangean Relaxation**

If all the constraints of a SCP are relaxed in a normal lagrangean relaxation fashion ([22]), one obtains the problem

$$(\mathrm{LSCP}_\lambda) \quad \min \ \Sigma_{j \varepsilon N} (c_j - \Sigma_{i \varepsilon M_j} \lambda_i)\, x_j + \Sigma_{i \varepsilon M} \lambda_i$$

$$x_j \varepsilon \{0,1\} \qquad (j \varepsilon N)$$

where $\lambda_i \geqslant 0$ is the multiplier associated with row i.

The optimal solution for LSCP_λ is

$$x_j = \begin{cases} 1 \text{ if } c_j - \Sigma_{i \varepsilon M_j} \lambda_i < 0 \\ 0 \text{ otherwise} \end{cases} \qquad (2.3)$$

and $v(\mathrm{LSCP}_\lambda)$ is a lower bound on v(SCP). Subgradient optimization can be used to modify the multipliers λ_i in order to improve the lagrangean lower bound, the best such bound being given by

$$(\mathrm{DLSCP}) \quad \max_{\lambda \geqslant 0} \quad v(\mathrm{LSCP}_\lambda)$$

It is well known that the optimal multipliers for DLSCP are equal to the optimal solution for DLP and v(DLSCP)=v(DLP).

If for a particular set of multipliers (λ_i) expression (2.1) is used replacing u_i by the corresponding λ_i, then a reduced cost is obtained for the variable x_j and it can be dropped from the SCP if

$$s_j + v(\mathrm{LSCP}_\lambda) \geqslant z_u \qquad (2.4)$$

2.5 **Combining LP and Lagrangean Relaxation**

Suppose that a number of subgradient optimization iterations are performed and that at each iteration we do:

(i) Remove any variable satisfying (2.4);

(ii) Starting with $N_0 = \Phi$ add to N_0 any variable for which the reduced cost became negative at any stage of the subgradient iterations.

(iii) Include in N_0 the variables in the best available feasible solution.

The set N_0 is then a "good" candidate subset of columns amongst which we will look for the solution to the SCP and will be used instead of the complete set N, for the LP relaxation of the SCP. Then, the restricted linear program which we denote by LP_0 gives a value $v(LP_0)$ greater than or equal to v(LP) (and could also be greater than v(SCP)). Nevertheless, if $v(LP_0)$ is close to v(LP) then the optimal dual variable to LP_0 can be used as a good approximation of the optimal dual variables to LP. Hence, the lagrangean multipliers in LSCP_λ may be initialised to those values and the lagrangean lower bound improved to a value close to v(DLP). The relation between $v(LP_o)$ is stated in the following proposition where DLP_o designates the dual linear problem of LP_o.

Proposition 2.1:

If $u^o = (u_i^o)_{i \in M}$ is an optimal solution to DLP_o, then:

$$v(DLP_o) + \Delta_o \leq v(LP) \leq v(LP_o) \tag{2.5}$$

where $\Delta_o = \Sigma_{j \in N-N_o} \min(0, c_j - \Sigma_{i \in M_j} u_i^o)$

The proof for this proposition follows easily from setting $\lambda_i = u_i^o$ in $LSCP_\lambda$ ([34]).

2.6 Computational Results of Reductions

The above procedure was computationally tested for three different class of problems depending on their cost function.

(I) costs c_j randomly generated from the interval [1,99]

(II) costs $c_j=1$

(III) costs c_j proportional to the cardinality of M_j:

i) $c_j = |M_j|$

ii) $c_j = 3.0 + |M_j|$

In class (I), two sets of test problems were considered:

(I.1) number of rows m=200; number of columns n=2000; density d=5%

(I.2) number of rows m=300; number of columns n=3000; density d=2%

For classes (II) and (III) the same set of problems was tested differing only in the costs. These problems were of size m=50, n=500, d=20%.

Table 1 summarises the computational results for the test problems in class (I). These are identified in column (1) by a general designation Tmxk where m is the number of rows and k is the number given to the test problem. The initial dimensions of the problem are shown in columns (2) and (3). Columns (4) and (5) in Table 1 give the dimensions of the test problem after doing the preliminary reductions. As expected for problems with random costs, the reduction on the number of variables is very significant mainly due to column dominance tests. For all test problems in Table 1 the number of variables was reduced by more than 80%. However, the resulting reduced problem is now a hard problem with the range of the costs much tighter.

Column (6) in Table 1 shows the values of the greedy heuristic upper bound. The bounds are reductions related to the first set of iterations for the Lagrangean relaxation as shown in columns (7) to (10). The upper bound (column 7) was improved for six of the test problems in Table 1. The lower bound is given in column (8), but its use did not imply any significant reduction in the dimensions of the problem as can be seen in columns (9) and (10).

Column (11) to (14) in Table 1 are related to the restricted linear program LP_o. The value $v(LP_o)$ is shown in column (12) while the corresponding lower bound to the SCP, $v(LP_o)+\Delta_o$, is given in column (12). This lower bound is for all test problems better than the previous value (column 8), and further reduction in the dimensions is achieved for most of the problems using reduced cost analysis. It is worthwhile to note here that the size of N_o was between 1/3 and 1/2 the size of N.

Finally, columns (15) to (18) show the outcome of the second phase application of lagrangean relaxation using as starting values of λ_i the optimal duals for LP_o. The lower bound (column 16) improves even further for all problems and an improvement on the upper bound over the value derived from greedy for almost all test problems. As a result the dimension of the problems is reduced even further and for test problem T300x1 the optimal value is obtained. It has been noted that the reduced size test problems still have approximately the same density as the original problem.

Table 2 shows the computational times of each step of the procedure for the test problems. The LP was solved using the XMP code developed by Marsten ([31]).

The computational results relative to the test problems of classes II and III are presented in Table 3 which has the same column designation as Table 1. We only show the computational results for a small set of test problems since the performance of the techniques described above is very similar for the different problems generated in those classes. As expected the preliminary reductions are not efficient for this type of problem and the other methods applied do not perform well either. For the proportional cost problems the LP procedure took too long (a limit of 100 seconds in the CDC 6500 was used) and, hence, the procedure is not completed for those problems.

2.7 Conclusion

In this section we presented a procedure to perform reductions in the dimensions of large size random SCP's. The conclusions are as follows:

i) For unicost and proportional-cost SCP's no significant reduction in the dimensions of the problems could be achieved;

ii) For random-cost SCP's, large problems (with nx10*m) are reduced to problems with $n \cong m$ of the same density as the original ones, and lower and upper bounds are obtained with a gap of less than 6% between them.

3. STATE SPACE RELAXATION FOR THE SCP

Dynamic programming can be used to solve the SCP but this requires, even for small size problems, too much storage and time to be useful in practice. One way of reducing the dimension of the state space of a dynamic program associated with the SCP, is presented in this section. Instead of obtaining an optimal solution to the problem, a lower bound is computed by solving dynamic programming recursions in a smaller state space. This corresponds to an idea developed and called state space relaxation (SSR) in [13] where it is used for the vehicle routing problem.

TABLE 1

Bounds and reductions produced by the procedure for large scale SCP's

PROBLEM	INITIAL DIMENSIONS		PRELIMINARY REDUCTIONS		GREEDY HEURISTICS	LAGRANGEAN RELAXATION (I)				LINEAR PROGRAMMING (RESTRICTED)				LAGRANGEAN RELAXATION (II)			
	m	n	m	n	z_u	z_u	z_ℓ	m	n	$\bar{z}_p$	z_ℓ	m	n	z_u	z_ℓ	m	n
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)
T200X1	200	2000	200	349	96.0	94.0	65.67	200	344	87.60	86.84	200	154	92.0	87.12	200	146
T200X2	200	2000	200	263	77.0	77.0	53.07	200	263	67.47	64.35	200	245	74.0	66.35	199	178
T200X3	200	2000	200	381	104.0	103.0	78.53	200	380	91.66	87.84	200	354	96.0	90.72	200	176
T200X4	200	2000	200	290	74.0	74.0	54.73	200	290	69.85	69.19	200	155	74.0	69.35	199	150
T200X5	200	2000	200	281	60.0	60.0	43.19	200	279	58.25	44.53	180	278	60.0	56.16	186	125
T300X1	300	3000	300	325	227.0	227.0	188.08	299	325	215.0	214.0	299	111	215.0	215.0	-	-
T300X2	300	3000	300	307	152.0	150.0	115.05	300	307	140.3	136.79	300	306	147.0	138.4	300	287
T300X3	300	3000	300	348	228.0	223.0	200.34	300	348	217.0	184.53	300	346	223.0	211.8	300	333
T300X4	300	3000	300	482	277.0	263.0	209.50	300	482	245.5	238.8	300	477	258.0	243.0	300	444
T300 5	300	3000	300	324	203.0	199.0	161.61	291	323	192.0	184.6	291	292	192.0	187.7	250	224

TABLE 2

Computing times for the procedure
(CDC 7600 seconds; FTN compiler)

PROBLEM (1)	PRELIMINARY REDUCTIONS (2)	GREEDY HEURISTIC (3)	LAGRANGEAN RELAXATION (4)	LINEAR PROGRAMMING (5)	LAGRANGEAN RELAXATION (6)	TOTAL (7)
T200X1	.49	.35	.94	4.22	.81	6.81
T200X2	.48	.33	1.04	4.30	.93	7.08
T200X3	.48	.36	1.05	5.40	.99	8.28
T200X4	.48	.33	1.91	3.17	.71	6.60
T200X5	.49	.31	.86	2.87	.75	5.28
T300X1	.45	.59	1.42	7.08	.52	10.06
T300X2	.45	.48	1.34	6.33	1.03	9.63
T300X3	.45	.50	1.52	6.90	1.50	10.87
T300X4	.45	.40	1.39	10.16	1.27	13.67
T300X5	.45	.54	1.29	4.20	1.05	7.53