European Journal of Operational Research 58 (1992) 293–300
North-Holland

Enhancing an algorithm for set covering problems

J.E. Beasley

The Management School, Imperial College, London SW7 2AZ, England

K. Jørnsten

Institute of Finance and Management Science, Norwegian School of Economics and Business Administration, Helleveien 30, N-5035 Bergen-Sandviken, Norway

Received January 1990; revised November 1990

Abstract: In this note we present enhancements to a previously published algorithm for the optimal solution of set covering problems. These enhancements relate to the use of a Lagrangean heuristic, feasible solution exclusion constraints, Gomory f-cuts and an improved branching strategy. Computational results, for problems involving up to 400 rows and 4000 columns, indicate that the enhanced algorithm gives better computational results than the original algorithm.

Keywords: Set covering, Lagrangean heuristic, f-cuts

1. Introduction

In this note we consider the set covering problem (SCP) which is the problem of covering the rows of an m-row, n-column, zero–one matrix (a_{ij}) by a subset of the columns at minimum cost. Defining

$$x_j = \begin{cases} 1 & \text{if column } j \text{ (cost } c_j > 0\text{) is in the solution,} \\ 0 & \text{otherwise,} \end{cases}$$

the SCP is:

$$\min \quad \sum_{j=1}^{n} c_j x_j \tag{1}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} a_{ij} x_j \geqslant 1, \quad i = 1,\ldots,m, \tag{2}$$

$$x_j \in (0, 1), \quad j = 1,\ldots,n. \tag{3}$$

Equation (2) ensures that each row is covered by at least one column and (3) is the integrality constraint.

The SCP, and variants of the SCP (such as the set partitioning problem (SPP) obtained by replacing the inequality in (2) by an equality), are important practical problems. They have applications in crew scheduling [2,3,8,21,31,32,35,38,39, 44,46], location of emergency facilities [36,43,47, 51], steel production [48–50], vehicle routing [7,27], ship scheduling [17,26], network attack or defence [13,14], assembly line balancing [28,41], traffic assignment in satellite communication systems [37], simplifying boolean expressions [16], the calculation of bounds in integer programs [18], information retrieval [20] and political districting [29].

In a previous paper [9] we presented an algorithm for the optimal solution of the SCP and in a recent paper [11] a Lagrangean heuristic for the heuristic solution of SCP's. In order to save space, therefore, we only survey here relevant work additional to that surveyed in [9] and [11].

1.1. Literature survey

Feo and Resende [24] presented a probabilistic heuristic for solving the SCP's that arise from Steiner triple systems.

Balas and Ng [5] presented a paper characterising all the facets of the set covering polytope defined by inequalities of the form $\sum_{j=1}^{n}\alpha_j x_j \geqslant 2$ where $\alpha_j \in \{0, 1, 2\}$, $j = 1, \ldots, n$. In a further paper Balas and Ng [6] showed that such facets can be obtained by the lifting of certain inequalities containing only three non-zero coefficients. No computational results were given in either [5] or [6].

Cornuejols and Sassano [19] and Sassano [42] presented papers characterising the facets of the set covering polytope defined by inequalities of the form $\sum_{j=1}^{n}\alpha_j x_j \geqslant \beta$ where $\alpha_j \in \{0, 1\}$; $j = 1, \ldots, n$, and β is a positive integer. Their approach is based on regarding (a_{ij}) as the adjacency matrix of a bipartite graph. No computational results were given in [19] and only two problems (based on Steiner triple systems) were solved in [42].

El-Darzi and Mitra [23] presented a paper detailing a variety of SCP's and SPP's that are publically available. Beasley [12] presented a paper detailing a variety of SCP's that are publically available via electronic mail.

Fisher and Kedia [25] presented an optimal algorithm based on dual heuristics for the mixed SCP/SPP problem (some constraints (cf. (2)) inequalities, some equalities). Computational results were given for the SCP for a number of test problems taken from the literature.

1.2. Algorithm overview

As noted above, in a previous paper [9] we presented an algorithm for the optimal solution of the SCP and in a recent paper [11] a Lagrangean heuristic for the heuristic solution of SCP's. In this note we enhance the algorithm given in [9] for the optimal solution of the SCP by:

(a) incorporating into it the Lagrangean heuristic for SCP's presented in [11];

(b) adding feasible solution exclusion constraints;

(c) using Gomory f-cuts; and

(d) using an improved branching strategy.

By solving test problems taken from the literature we are able to compare the algorithm given in this paper with algorithms [4,9,25] given by other authors.

In order to save space we need to assume throughout this note some familiarity with both [9] and [11]. However to aid the reader we give below a brief overview of [9] and [11]. Essentially the algorithm given in [9] consists of:

(a) A dual ascent procedure for the linear programming (LP) relaxation of the SCP (replace (3) by $x_j \geqslant 0$).

(b) A Lagrangean relaxation of (2). If $s_i \geqslant 0$, $i = 1, \ldots, m$, are Lagrange multipliers for (2), the associated Lagrangean relaxation is easily solved to give a solution (X_j) together with a lower bound Z_{LB} on the optimal solution to the original SCP.

(c) A subgradient ascent procedure. At each subgradient iteration, if Z_{max} is the maximum lower bound found and Z_{UB} the best feasible solution found:

1) Solve the Lagrangean relaxation with the current set of multipliers (s_i) to get the solution Z_{LB}, (X_j). Update Z_{max} by $Z_{max} = \max(Z_{max}, Z_{LB})$.

2) Calculate the subgradients (G_i) using

$$G_i = 1 - \sum_{j=1}^{n} a_{ij} X_j, \quad i = 1, \ldots, m. \tag{4}$$

3) Define a step size T by

$$T = \frac{f(1.05 Z_{UB} - Z_{LB})}{\sum_{i=1}^{m} (G_i)^2} \tag{5}$$

where $f = 2$ initially. In the computational results reported below, if Z_{max} had not increased in the last 50 iterations with the current value of f, then f was halved.

4) Update the Lagrange multipliers using

$$s_i = \max(0, s_i + TG_i), \quad i = 1, \ldots, m. \tag{6}$$

d) Optimally solving the LP relaxation of the SCP.

e) Resolving the SCP by a tree search procedure with bounds being calculated by Lagrangean relaxation and the subgradient ascent procedure.

A comprehensive set of problem reduction tests are also applied at various stages in the algorithm. The details can be found in [9].

The Lagrangean heuristic given in [11] essentially, at each subgradient iteration, adjusts (X_j) into a feasible solution for the SCP and updates Z_{UB} accordingly. The details can be found in [11].

2. Algorithm enhancements

2.1. Lagrangean heuristic

For the algorithm given in [9] we used the heuristic of Balas and Ho [4] in order to generate feasible solutions for the SCP (upper bounds upon the optimal solution of the SCP). Computational results given in [11] indicate that, for non-unicost SCP's ($c_j \neq c_k$ for some $j \neq k$), the Lagrangean heuristic given in that paper is a better heuristic for the SCP than the Balas and Ho heuristic. Hence our first enhancement to the algorithm presented in [9] is to use the Lagrangean heuristic in place of the Balas and Ho heuristic for non-unicost problems.

2.2. Feasible solution exclusion constraints

Let S be a set of columns corresponding to the best feasible solution for the SCP found at the end of the subgradient ascent procedure (just before optimally solving the LP relaxation of the SCP) where, without loss of generality, we assume that S-$[j]$ ($j \in S$) is not a feasible solution for the SCP. Then we can add to the SCP the two feasible solution exclusion constraints:

$$\sum_{j \in S} x_j \leqslant |S| - 1, \tag{7}$$

$$\sum_{j \notin S} x_j \geqslant 1. \tag{8}$$

These constraints imply that, for an improved feasible solution, at least one column currently in S must be replaced by at least one column currently not in S. Adding these constraints to the SCP excludes S from the solution to that problem ((1)–(3),(7),(8)) and so makes that problem infeasible if S is the unique optimal solution. Such feasible solution exclusion constraints have previously been used by us in [1,10].

2.3. Gomory *f*-cuts

After solution of the LP relaxation of the SCP with feasible solution exclusion constraints added we used Gomory *f*-cuts in order to increase the lower bound on the optimal solution to the SCP obtained from the LP.

Limited computational experience indicated that a good strategy was to:

(a) generate 30 Gomory *f*-cuts from the LP solution and then resolve the LP using a dual simplex algorithm;

(b) generate half these *f*-cuts from the x_j variables with the largest fractional values in the optimal solution to the LP; and

(c) generate the remaining *f*-cuts from the slack variables (for the constraints (2),(7),(8)) with the largest fractional values in the optimal solution to the LP.

Details of how to calculate Gomory *f*-cuts from an optimal LP simplex tableau can be found in most integer programming textbooks (e.g. [45]).

Note here that the 30 Gomory *f*-cuts and the two feasible solution exclusion constraints can be regarded as being of the form

$$\sum_{j=1}^{n} a_{ij} x_j \geqslant \pm 1, \tag{9}$$

i.e. similar to the SCP constraints (2), where the a_{ij} are now fractional (and positive or negative). They can therefore be incorporated into the Lagrangean relaxation and subgradient ascent procedure in virtually the same manner as the SCP constraints.

The details of this we leave to the reader but note here that initial Lagrange multipliers for these constraints at the start of the tree search are provided by the dual variables associated with these constraints in the LP solution found after addition of the feasible solution exclusion constraints and the 30 Gomory *f*-cuts.

We would comment here that the use of cuts in solution algorithms for the SCP has been a frequent theme in the literature (e.g. see [4,15,30,33,34,40] as well as the more recent theoretical work [5,6,19,42] on facets of the set covering polytope).

2.4. Branching strategy

In forward branching from each tree node we:

(a) apply the first pass of the dual ascent procedure given in [9] to the current set of Lagrange multipliers;

(b) choose the (uncovered) row i with the maximum value of $|s_i G_i|$;

(c) branch by choosing the column j (from those which cover row i and for which $X_j = 1$) with the maximum value in the LP solution obtained after addition of the Gomory f-cuts (ties broken arbitrarily).

3. Computational results

The algorithm presented in this note was programmed in FORTRAN and run on a Cray X-MP/28 using the CFT compiler (with maximum optimisation) for the same set of test problems as were solved in [9]. Details of this set of test problems are given in Table 1. Note here that all of these test problems are now publically available via electronic mail from OR-Library [12].

Table 2 gives the results for problem sets 4–6 and Table 3 the results for problem sets A–E. In Table 4 we give a comparison between the enhanced algorithm presented in this note, the original algorithm [9], the algorithm of Fisher and Kedia [25] and the algorithm of Balas and Ho [4].

Comparing the algorithm presented in this note with the algorithm presented by Fisher and Kedia [25] on problem sets 4–6 (comprising 25 problems in all) we have, from Tables 2–4 and the corresponding tables in [25], that the algorithm presented in this note:

(a) solves 18 problems without branching being necessary, as opposed to only 7 problems for the algorithm presented in [25];

(b) solves all 25 problems in fewer (or the same number of) tree nodes than the algorithm presented in [25];

(c) requires (in total) 84% fewer tree nodes than the algorithm presented in [25]; and

(d) is computationally inferior to the algorithm presented in [25] if the Cray X-MP/28 is faster than the DEC 10 by a factor of $[(10*33.7 + 10*77.1 + 5*1098.5)/(10*1.4 + 10*2.4 + 5*6.4)] = 94.3$.

Figures given in Dongarra [22] suggest that the Cray X-MP/28 is faster than the DEC 10 by this factor so we can conclude that the algorithm given in this note is computationally inferior to the algorithm of Fisher and Kedia [25].

Whether the Fisher and Kedia [25] algorithm retains its computational advantage for the much larger problems (problem sets A–D) solved in this note is currently an open question.

Comparing the enhanced algorithm presented here with the original algorithm [9] on problem sets 4–6 and A–E (comprising 50 problems in all) we have, from Tables 2–4 and the corresponding tables in [9], that the algorithm presented in this note:

(a) solves 18 problems without branching being necessary, as opposed to only 13 problems for the algorithm presented in [9];

(b) solves all but six problems in fewer (or the same number of) tree nodes than the algorithm presented in [9];

(c) requires (in total) 41% fewer tree nodes than the algorithm presented in [9]; and

(d) is computationally inferior to the algorithm presented in [9] if the Cray X-MP/28 is faster than the Cray-1S by a factor of $[(10*3.7 + 10*6.8 + 5*21.2 + 5*70.6 + 5*256.6 + 5*519.3 + 5*1327.7 + 5*44.8)/(10*1.4 + 10*2.4 + 5*6.4 + 5*18.2 + 5*57.3 + 5*105.3 + 5*255.0 + 5*22.2)] = 4.8$.

Figures given in Dongarra [22] suggest that the Cray X-MP/28 is not faster than the Cray-1S by this factor so we can conclude that the enhanced algorithm presented in this note is computationally superior to the original algorithm [9].

Table 1
Test problem details

Problem set	Number of rows (m)	Number of columns (n)	Density [a]	Number of problems in problem set
4	200	1000	2%	10
5	200	2000	2%	10
6	200	1000	5%	5
A	300	3000	2%	5
B	300	3000	5%	5
C	400	4000	2%	5
D	400	4000	5%	5
E	50	500	20% (unicost)	5

[a] The density of an SCP is the percentage of ones in the (a_{ij}) matrix.

4. Conclusions

In this note we have presented enhancements to an algorithm for the optimal solution of set covering problems. Computational results indicated that the enhanced algorithm gave better-quality results than the original algorithm.

Table 2
Computational results – problem sets 4–6

Problem number	Setup/initial subgradient ascent/reduction						Linear programs/reduction					Optimal value	Number of tree nodes	Total time Cray X-MP/28 seconds
	Number of subgradient iterations	Rows	Columns	Maximum lower bound (Z_{max})	Best upper bound (Z_{UB})	Time Cray X-MP/28 seconds	LP solution value before *f*-cuts	LP solution value after *f*-cuts	Rows	Columns	Time Cray X-MP/28 seconds			
4.1	483	–	–	429	429	0.92	–	–	–	–	–	429	–	0.92
4.2	310	–	–	512	512	0.89	–	–	–	–	–	512	–	0.89
4.3	291	–	–	516	516	0.90	–	–	–	–	–	516	–	0.90
4.4	1062	13	15	493.98	495	1.44	494	494	–	–	0.03	494	–	1.47
4.5	360	–	–	512	512	0.89	–	–	–	–	–	512	–	0.89
4.6	1197	86	74	557.21	561	2.25	558.50	560	–	–	0.05	560	–	2.30
4.7	452	–	–	430	430	0.95	–	–	–	–	–	430	–	0.95
4.8	1167	67	59	488.64	492	2.02	488.67	490.50	–	–	0.05	492	–	2.07
4.9	1199	39	37	638.39	641	2.23	640	641	–	–	0.03	641	–	2.26
4.10	501	–	–	514	514	0.97	–	–	–	–	–	514	–	0.97
5.1	1235	153	130	251.19	255	3.68	251.23	252	–	–	0.70	253	–	4.38
5.2	1420	179	151	299.65	303	4.28	299.76	300.91	166	126	1.60	302	11	6.05
5.3	196	–	–	226	226	1.17	–	–	–	–	–	226	–	1.17
5.4	984	–	–	242	242	1.89	–	–	–	–	–	242	–	1.89
5.5	324	–	–	211	211	1.24	–	–	–	–	–	211	–	1.24
5.6	406	–	–	213	213	1.32	–	–	–	–	–	213	–	1.32
5.7	1146	69	68	291.75	294	2.57	292.11	293	55	49	0.23	293	5	2.84
5.8	1201	–	–	288	288	2.29	–	–	–	–	–	288	–	2.29
5.9	511	–	–	279	279	1.51	–	–	–	–	–	279	–	1.51
5.10	251	–	–	265	265	1.21	–	–	–	–	–	265	–	1.21
6.1	1423	199	136	133.12	140	4.62	133.14	133.52	198	131	2.28	138	41	7.81
6.2	1253	198	113	140.40	146	3.83	140.46	141.24	195	103	1.74	146	67	6.99
6.3	1405	173	93	139.98	145	3.65	141.72	142.14	140	70	0.98	145	33	5.16
6.4	953	117	65	128.91	131	2.30	129	129	80	53	0.43	131	5	2.79
6.5	1164	191	139	153.21	162	4.37	153.35	154.02	189	130	2.17	161	129	9.17

Table 3
Computational results – problem sets A–E

Problem number	Setup/initial subgradient ascent/reduction						Linear programs/reduction					Optimal value	Number of tree nodes	Total time Cray X-MP/28 seconds
	Number of subgradient iterations	Rows	Columns	Maximum lower bound (Z_{max})	Best upper bound (Z_{UB})	Time Cray X-MP/28 seconds	LP solution value before *f*-cuts	LP solution value after *f*-cuts	Rows	Columns	Time Cray X-MP/28 seconds			
A.1	1217	296	292	246.75	256	9.71	246.84	247.28	287	279	6.94	253	373	28.26
A.2	1191	291	289	247.33	256	9.63	247.50	247.78	263	239	7.39	252	357	27.78
A.3	1536	253	226	227.90	233	8.15	228	228.23	251	219	5.26	232	377	24.12
A.4	1541	175	122	231.26	234	5.21	231.51	232.09	122	90	0.88	234	17	6.33
A.5	1369	128	105	234.87	237	4.09	235	235.27	96	77	0.53	236	15	4.74
B.1	1753	296	208	64.49	70	10.74	64.54	64.86	277	182	6.15	69	137	20.66
B.2	1482	300	294	69.27	77	13.56	69.30	69.41	300	284	10.31	76	1655	78.93
B.3	1110	300	251	74.11	81	11.23	74.16	74.42	300	235	7.79	80	671	38.74
B.4	1218	300	325	71.17	80	12.78	71.22	71.39	300	311	10.49	79	3157	125.27
B.5	1278	300	173	67.64	72	8.04	67.67	67.85	299	164	5.36	72	355	22.99
C.1	1136	393	347	223.74	231	16.17	223.80	224.19	390	327	16.40	227	549	51.92
C.2	1297	385	428	212.73	222	17.46	212.85	213.26	385	402	18.81	219	619	62.10
C.3	1370	385	478	234.49	246	19.69	234.58	234.82	385	469	20.52	243	5957	297.54
C.4	1404	400	390	213.73	222	18.97	213.85	214.26	377	365	14.78	219	515	53.15
C.5	1248	367	302	211.51	217	13.46	211.64	211.99	366	279	12.11	215	1001	61.67
D.1	1366	400	241	55.26	60	17.41	55.31	55.41	400	223	12.70	60	1015	67.59
D.2	1146	400	405	59.20	68	21.77	59.35	59.56	400	381	20.50	66	7179	335.92
D.3	1393	400	416	65.01	74	24.22	65.07	65.24	400	399	23.16	72	10455	471.95
D.4	1681	400	356	55.76	63	20.90	55.84	56	400	326	16.40	62	8687	376.78
D.5	1401	399	157	58.59	61	12.63	58.62	58.77	399	138	7.86	61	63	22.72
E.1	1247	50	198	3.47	5	16.96	3.48	3.49	50	107	1.53	5	39	19.06
E.2	1121	50	371	3.38	5	18.04	3.38	3.39	50	268	4.32	5	109	24.54
E.3	1155	50	457	3.29	5	19.46	3.30	3.32	50	343	6.04	5	131	28.43
E.4	906	50	303	3.45	5	14.28	3.45	3.48	50	113	2.07	5	59	17.20
E.5	964	50	374	3.39	5	16.12	3.39	3.41	50	216	3.54	5	127	21.89

Table 4
Algorithm comparison

	Problem set	Enhanced algorithm	Beasley [9] algorithm	Fisher and Kedia [25] algorithm	Balas and Ho [4] [a] algorithm
Number of problems solved to optimality	4	10	10	10	10
	5	10	10	10	9
	6	5	5	5	2
	A–E	5	5		
Number solved to optimality without tree search	4	10	7	5	6
	5	8	6	2	4
	6	0	0	0	0
	A–E	0	0		
Average number of tree nodes (excluding initial tree node)	4	0	2.2	10.3	9.1
	5	1.4	13.8	25.6	17.4
	6	54.0	130.0	284.2	131.2
	A	226.8	463.6		
	B	1194.0	2020.8		
	C	1727.2	3632.0		
	D	5478.8	8474.0		
	E	92.0	147.2		
Average computer time (seconds)	4	1.4	3.7	33.7	111.3
	5	2.4	6.8	77.1	207.6
	6	6.4	21.2	1098.5	1428.4
	A	18.2	70.6		
	B	57.3	256.6		
	C	105.3	519.3		
	D	255.0	1327.7		
	E	22.2	44.8		
Computer used		Cray X-MP/28	Cray-1S	DEC 10	DEC 20/50

[a] For the Balas and Ho [4] algorithm the figures given for the average number of tree nodes and for the average computer time include problems not solved to optimality due to time limit restrictions.

References

[1] Aboudi, R., and Jørnsten, K., "Some extensions to the pivot and complement heuristic for 0–1 integer programming problem", Working Paper, available from the second author at Institute of Finance and Management Science, Norwegian School of Economics and Business Administration, Helleveien 30, N-5035 Bergen-Sandviken, Norway, 1988.

[2] Ali, A.I., and Thiagarajan, H., "A network relaxation based enumeration algorithm for set partitioning", *European Journal of Operational Research* 38 (1989) 76–85.

[3] Baker, E.K., Bodin, L.D., Finnegan, W.F., and Ponder, R.J., "Efficient heuristic solutions to an airline crew scheduling problem", *AIIE Transactions* 11 (1979) 79–85.

[4] Balas, E., and Ho, A., "Set covering algorithms using cutting planes, heuristics, and subgradient optimization: a computational study", *Mathematical Programming* 12 (1980) 37–60.

[5] Balas, E., and Ng, S.M., "On the set covering polytope: I. All the facets with coefficients in {0, 1, 2}", *Mathematical Programming* 43 (1989) 57–69.

[6] Balas, E., and Ng, S.M., "On the set covering polytope: II. Lifting the facets with coefficients in {0, 1, 2}", *Mathematical Programming* 45 (1989) 1–20.

[7] Balinski, M.L., and Quandt, R.E., "On an integer program for a delivery problem", *Operations Research* 12 (1964) 300–304.

[8] Bartholdi, J.J., "A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering", *Operations Research* 29 (1981) 501–510.

[9] Beasley, J.E., "An algorithm for set covering problems", *European Journal of Operational Research* 31 (1987) 85–93.

[10] Beasley, J.E., "An algorithm for solving large capacitated warehouse location problems", *European Journal of Operational Research* 33 (1988) 314–325.

[11] Beasley, J.E., "A Lagrangian heuristic for set-covering problems", *Naval Research Logistics* 37 (1990) 151–164.

[12] Beasley, J.E., "OR-Library: Distributing test problems by electronic mail", *Journal of the Operational Research Society* 41 (1990) 1069–1072.

[13] Bellmore, M., Greenberg, H.J., and Jarvis, J.J., "Multi-commodity disconnecting sets", *Management Science* 16 (1970) B427–B433.

[14] Bellmore, M., and Ratliff, H.D., "Optimal defence of

multi-commodity networks", *Management Science* 18 (1971) B174–B185.

[15] Bellmore, M., and Ratliff, H.D., "Set covering and involutory bases", *Management Science* 18 (1971) 194–206.

[16] Breuer, M.A., "Simplification of the covering problem with application to boolean expressions", *Journal for the Association of Computing Machinery* 17 (1970) 166–181.

[17] Brown, G.G., Graves, G.W., and Ronen, D., "Scheduling ocean transportation of crude oil", *Management Science* 33 (1987) 335–346.

[18] Christofides, N., "Zero–one programming using non-binary tree search", *The Computer Journal* 14 (1971) 418–421.

[19] Cornuejols, G., and Sassano, A., "On the 0,1 facets of the set covering polytope", *Mathematical Programming* 43 (1989) 45–55.

[20] Day, R.H., "On optimal extracting from a multiple file data storage system: An application of integer programming", *Operations Research* 13 (1965) 482–494.

[21] Desrochers, M., and Soumis, F., "A column generation approach to the urban transit crew scheduling problem", *Transportation Science* 23 (1989) 1–13.

[22] Dongarra, J.J., "Performance of various computers using standard linear equations software in a Fortran environment", *ACM Computer Architecture News* 13 (1985) 3–9.

[23] El-Darzi, E., and Mitra, G., "Set covering and set partitioning: A collection of test problems", *OMEGA* 18 (1990) 195–201.

[24] Feo, T.A., and Resende, M.G.C., "A probabilistic heuristic for a computationally difficult set covering problem". *Operations Research Letters* 8 (1989) 67–71.

[25] Fisher, M.L., and Kedia, P., "Optimal solution of set covering/partitioning problems using dual heuristics", *Management Science* 36 (1990) 674–688.

[26] Fisher, M.L., and Rosenwein, M.B., "An interactive optimisation system for bulk-cargo ship scheduling", *Naval Research Logistics* 36 (1989) 27–42.

[27] Foster, B.A., and Ryan, D.M., "An integer programming approach to the vehicle scheduling problem", *Operational Research Quarterly* 27 (1976) 367–384.

[28] Freeman, D.R., and Jucker, J.V., "The line balancing problem", *Journal of Industrial Engineering* 18 (1967) 361–364.

[29] Garfinkel, R.S., and Nemhauser, G.L., "Optimal political districting by implicit enumeration techniques", *Management Science* 16 (1970) B495–B508.

[30] House, R.W., Nelson, L.D., and Rado, T., "Computer studies of a certain class of linear integer problems", in: *Recent Advances in Optimization Techniques*, A. Lavi, and T.P. Vogel (eds.), John Wiley, New York, 1966, 241–280.

[31] Marsten, R.E., Muller, M.R., and Killion, C.L., "Crew planning at Flying Tiger: A successful application of integer programming", *Management Science* 25 (1979) 1175–1183.

[32] Marsten, R.E., and Shepardson, F., "Exact solutions of crew scheduling problems using the set partitioning problem: Recent successful applications", *Networks* 11 (1981) 165–177.

[33] Martin, G.T., "An accelerated Euclidean algorithm for integer linear programming", in: *Recent Advances in Mathematical Programming*, R.L. Graves and P. Wolfe (eds.), McGraw-Hill, New York, 1963, 311–317.

[34] Martin, G.T., "Integer programming: Gomory plus ten", Paper presented at ORSA Meeting in Miami, November, 1969. Abstract given in *Bulletin of the Operations Research Society of America* 17 (1969) B-222.

[35] Parker, M., and Smith, B., "Two approaches to computer crew scheduling", in: A. Wren (ed.), *Computer Scheduling of Public Transport: Urban Passenger Vehicle and Crew Scheduling*, North-Holland, Amsterdam, 1981, 193–222.

[36] Revelle, C., Marks, D., and Liebman, J.C., "An analysis of private and public sector location models", *Management Science* 16 (1970) 692–707.

[37] Ribeiro, C.C., Minoux, M., and Penna, M.C., "An optimal column-generation-with-ranking algorithm for very large-scale set partitioning problems in traffic assignment", *European Journal of Operational Research* 41 (1989) 232–239.

[38] Rubin, J., "A technique for the solution of massive set covering problems with application to airline crew scheduling", *Transportation Science* 7 (1973) 34–48.

[39] Ryan, D.M., and Falkner, J.C., "On the integer properties of scheduling set partitioning models", *European Journal of Operational Research* 35 (1988) 442–456.

[40] Salkin, H.M., and Koncal, R.D., "Set covering by an all integer algorithm: Computational experience", *Journal of the Association for Computing Machinery* 20 (1973) 189–193.

[41] Salveson, M.E., "The assembly line balancing problem", *Journal of Industrial Engineering* 6 (1955) 18–25.

[42] Sassano, A., "On the facial structure of the set covering polytope", *Mathematical Programming* 44 (1989) 181–202.

[43] Schreuder, J.A.M., "Application of a location model to fire stations in Rotterdam", *European Journal of Operational Research* 6 (1981) 212–219.

[44] Shepardson, F., and Marsten, R.E., "A Lagrangean relaxation algorithm for the two duty period scheduling problem", *Management Science* 26 (1980) 274–281.

[45] Taha, H.A., *Integer Programming: Theory, Applications and Computations*, Academic Press, London, 1975, 179.

[46] Tibrewala, R., Phillipe, D., and Browne, J., "Optimal scheduling of two consecutive idle periods", *Management Science* 19 (1972) 71–75.

[47] Toregas, C., Swain, R., Revelle, C., and Bergman, L., "The location of emergency service facilities", *Operations Research* 19 (1971) 1363–1373.

[48] Vasko, F.J., Wolf, F.E., and Stott, K.L., "Optimal selection of ingot sizes via set covering", *Operations Research* 35 (1987) 346–353.

[49] Vasko, F.J., Wolf, F.E., and Stott, K.L., "A set covering approach to metallurgical grade assignment", *European Journal of Operational Research* 38 (1989) 27–34.

[50] Vasko, F.J., Wolf, F.E., Stott, K.L., and Scheirer, J.W., "Selecting optimal ingot sizes for Bethlehem Steel", *Interfaces* 19/1 (1989) 68–84.

[51] Walker, W., "Using the set-covering problem to assign fire companies to fire houses", *Operations Research* 22 (1974) 275–277.