Available online at http://scik.org
J. Math. Comput. Sci. 5 (2015), No. 2, 237-245
ISSN: 1927-5307

A NEW RELAXATION FOR THE SET PROBLEMS

FARHAD DJANNATY[1,*], MUHAMMAD YARALI[2]

[1]Soran University, Soran, Iraq

[2]Payame Noor, Tehran, Iran

Abstract. Set problems (SP) are an important class of combinatorial optimization problems which have many practical applications. Network relaxations of SP are alternative ways of relaxing the problem to find quick lower bound on the value of the objective function. Inspired by these relaxations, a new simple and much faster relaxation of the set problems is proposed. Using a standard cost allocation strategy and an innovative, the new relaxation is applied to a number of standard SP test problems and computational results are presented.

Keywords: network relaxation; set problem; shortest route relaxation.

2000 AMS Subject Classification: 44A55.

1 Introduction

Set problems (SP) comprising, set covering problem (SCP), set partitioning problem (SPP), and set packing problem (SPK) have many applications in bus, railway, airline crew scheduling, plant location, circuit switching, information retrieval, assembly line balancing, political districting, and truck delivery [3,9,10]. More recent applications of SCP are found in probe selection in hybridization experiments in DNA sequencing [7] and feature selection and pattern construction in logical analysis of data [8].

A number of procedures has been developed which can deal with set problems [5,7,9]. They used either cutting plane algorithm and/or branch and bound algorithm and then found that these algorithms are shown to have exponential and data dependent computing time. Nemhauser, G.L. [11]. Beasley [5] has developed a tree search method to solve the SCP. The continuous development of mathematical programming has much improved the performance of exact

[*]Corresponding author

Received November 29, 2014

branch-and-bound algorithms [3, 5, 7, 9] accompanying with advances in computational machinery. Recent exact branch-and-bound algorithms enable us to solve large SCP instances with up to 400 rows and 4000 columns exactly [3]. Afif and *et al* have developed a new heuristic based on the flow algorithm of Ford and Fulkerson. The set covering problem can be relaxed to form an assignment problem, a minimal spanning tree problem, a shortest route problem [10].
Heuristic algorithms have also been studied extensively [12], and several efficient metaheuristic algorithms have been developed to solve huge SCP instances with up to 5000 rows and 1,000,000 columns within about 1% of the optimum in a reasonable computing time [3, 4].
This paper is organized in 4 sections. In section 2 the mathematical model of the Set problems is explained. In section 3 the proposed relaxation is described and it is established that our relaxation is a proper one. In section 4 a numerical example is presented. In section 5 two cost allocation strategies are proposed. In section 6 a computational experiment is carried out which provides quick lower bounds for the set problem and the ending part is a concluding remark.

2. Model of the Set Problems

Let $M = \{1,2, \dots , m\}$ be the set of m integers and let S denote a set of n subsets of
M. Thus

$$N = \{1,2, \dots , n\}$$

$$S = \{s_1, s_2, \dots , s_n\} \quad \text{where} \quad s_i \subseteq M , i \in N$$

Let

$$a_{ij} = \begin{cases} 1 , & if \quad i \in s_j \\ 0 , & if \quad i \notin s_j \end{cases} \qquad (i = 1,2, \dots m, \; j = 1,2, \dots , n)$$

The set covering problem (SCP) can be defined as follows:

Minimize $\Sigma \, c_j x_j$

Subject to

$$\sum_{j=1}^{j=n} a_{ij} x_j \geq 1, \quad (i = 1, \dots , m),$$

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

Where decision variable x_j indicates whether x_j is selected or not and c_j is the cost associated with selecting s_j. The problem can be interpreted as finding the minimum cost selection of subsets of S such that each member of M is covered by at least one member of the selected subset of S. If we replace " $\geq$ " by " = " in each of the constraints of the above model, the modified problem is called the set partitioning problem (SPP). If " $\geq$ " is replaced by " $\leq$ " and the objective function is to be maximized, the resulting model is the set packing problem (SPK).

Set problems are classified as NP-complete [10], which means that no polynomial time algorithm is known that guarantees to solve every instance of these problems. This increases the importance of relaxations which yield sharp lower and upper bounds as quickly as possible to be used in a branch and bound algorithm.

3. A New Relaxation for the Set Problems

A number of network based relaxations is proposed for the set problems by El-Darzi [10] which can provide quick lower bounds for the set problems. In these relaxations the columns of the set covering problem are decomposed into a number of segments of ones based on which a graph is constructed and the column costs are distributed among the arcs created from the same parent column and a lower bound is computed using a network flow algorithm. Column decomposition, storing the resulting network, plus more effort and memory usage are . Two of these relaxations are time consuming and produce weak bounds [14]. There are some faults with the assignment relaxation of El-Darzi which leads to infeasible relaxation and thus no lower bound can be obtained in most of the test problems used in this paper. A small example of this infeasibility is provided in [14]. Our computational experience reveals the fact that segments of two and more nonzeros which are the only advantage of the shortest route relaxation over the proposed relaxation is not utilized in finding the shortest route and thus in computing lower bound[9]. Only in the last two test problems, arcs associated with segments containing more than one nonzero entry are contributing to the reduced network and thus contribute to the shortest route. Therefore, our proposed relaxation does not suffer from not considering these kinds of segments.

In the proposed relaxation the original data of the problem such as the column costs, the number of non-zeros in the column and the position of the nonzero are used and there is no need for the column decomposition. The following notations are adopted in the paper:

Let $R_i = \{ j \mid a_{ij} = 1, j = 1,2,\cdots,n \} \quad i = 1,2,\cdots,m$

$$H_j = \{\, i \mid a_{ij} = 1, i = 1,2,\cdots,m \,\} \quad j = 1,2,\cdots,n$$

Let associate with each non-zero entry a_{ij} in the A matrix a binary variable y_{ij} which is 1 if it's associated cost (which is determined during the cost allocation) is selected as the minimum cost of row i and is 0 otherwise. The cost associated with y_{ij} is denoted by d_{ij} subject to the following condition:

$$\sum_{i \in H_j} d_{ij} = c_j \quad j = 1,2,\cdots,n$$

The relaxed problem can be stated as follows:

$$Min \sum_{i=1}^{i=m} \sum_{j \in R_i} d_{ij}\, y_{ij}$$

$$\sum_{j \in R_i} y_{ij} \geq 1 \qquad i = 1,2\cdots,m$$

$$y_{ij} \in \{0,1\} \quad i \in H_j \ \ and \ \ j = 1,2,\dots,n$$

If $\geq$ is replaced by $=$ a new relaxation for the set partitioning problem will be resulted and if min is replaced by max and all cases of $\geq$ are replaced by $\leq$ a new upper bound for the set packing problem is obtained.

The optimal solution to the above problem can easily be found by taking the minimum over all d_{ij} associated with row i and set the corresponding binary variable y_{ij} equal to 1 and set the rest of y_{ij}'s in row i to 0. In other words let

$$D_i = min\{d_{ij} \mid j \in R_i\} \quad i = 1,2,\dots,m$$

Let $$y_{ij} = \left[\frac{D_i}{d_{ij}}\right] \quad i = 1,2,\dots,m \ \ and \ \ j \in R_i$$

Where $[x]$ is the greatest integer which is less than or equal to *x*. The reason for the optimality of the above solution is that it satisfies all the constraints and it produces the minimum objective function value.

$$\sum_{i=1}^{i=m} \sum_{j \in R_i} d_{ij}\, y_{ij} = \sum_{i=1}^{i=m} \sum_{j \in R_i} d_{ij} \left[\frac{D_i}{d_{ij}}\right] \geq \sum_{i=1}^{i=m} \sum_{j \in R_i} D_i \left[\frac{D_i}{d_{ij}}\right]$$

$$\sum_{i=1}^{i=m} D_i \sum_{j \in R_i} \left[\frac{D_i}{d_{ij}}\right] \geq \sum_{i=1}^{i=m} D_i.$$

Because at least once $D_i = d_{ij}$ for some $j \in R_i$. Therefore, $\sum_{i=1}^{i=m} D_i$ is a lower bound for both the set covering problem and the associated set partitioning problem. It is easy to show that this solution satisfies all the constraints, because

$$\sum_{j \in R_i} y_{ij} = \sum_{j \in R_i} \left[\frac{D_i}{d_{ij}}\right] \geq 1 \qquad i = 1,2\cdots,m$$

The above inequality holds because $D_i = d_{ij}$ occurs at least once when $j \in R_i$ for $i = 1,2 \cdots, m$. Therefore, $y_{ij} = 1$ happens at least once when $j \in R_i$.

In order to demonstrate that the proposed relaxation is a proper relaxation it is enough to show that an arbitrary solution of the set covering problem is included in the solution set of the proposed relaxation. The corresponding solution in the relaxation is as follows:

$$y_{ij} = x_j \quad \forall\, i \in R_j, \text{ and }, \quad j = 1, \cdots, n$$

Where x_j is a decision variable of the set problem.

4. Numerical Example

The proposed relaxation is applied to the following set covering or set partitioning problem which is designed by the authors and the lower bound is numerically calculated. The reduced cost vector is assumed to be,

[8 8 8 8 8 8 8 8 8 8 8 8 10 10 10 10 10 10 10 10 10 10 10 10 12 12 12 12 12 12]

And the following 15×30 matrix presents technological coefficients of the set problem:

1	0	0	0	1	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0	1	0	0	1	1	0	1	0	1	0	0	0	1	0	1	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1	0	0	0	1
1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	1	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	1	0	1
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	1	0	1	0	0	1	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0

H_js are successively found as follows:

$H_1 = \{1,4,8,13,15\}$ $H_2 = \{3,5,9\}$ $H_3 = \{2,10\}$ $H_4 = \{4,6,10,14\}$ $H_5 = \{1,8,13\}$ $H_6 = \{8,13\}$

$H_7 = \{7,14,15\}$ $H_8 = \{7,12\}$ $H_9 = \{2,3,4\}$ $H_{10} = \{1,6,8,15\}$ $H_{11} = \{11\}$ $H_{12} = \{1,4,12,15\}$
$H_{13} = \{4,5,12,13\}$ $H_{14} = \{1,12,14\}$ $H_{15} = \{4,13,14,15\}$ $H_{16} = \{1,2,7,9,12,15\}$ $H_{17} = \{3,4,9\}$
$H_{18} = \{1,8,11\}$ $H_{19} = \{13,14,15\}$ $H_{20} = \{1,2,3,7,12\}$ $H_{21} = \{4,8\}$ $H_{22} = \{1,2,3,11\}$ $H_{23} =$
$\{4,7\}$ $H_{24} = \{5,6,13\}$ $H_{25} = \{1,7,9,15\}$ $H_{26} = \{2,3,6,7\}$ $H_{27} = \{6,8,10\}$ $H_{28} = \{4,6,9\}$
$H_{29} = \{4,6,8,10\}$, $H_{30} = \{1,4,7,9,11\}$

R_i's are computed as follows:

$R_1 = \{1,5,10,12,14,16,18,20,22,25,30\}$ $R_2 = \{3,9,16,20,22,26\}$ $R_3 = \{2,9,17,20,22,26\}$
$R_4 = \{1,4,9,12,13,15,17,21,23,28,29,30\}$ $R_5 = \{2,13,24\}$ $R_6 = \{4,10,24,26,27,28,29\}$
$R_7 = \{7,8,16,20,23,25,26,30\}$ $R_8 = \{1,5,6,10,18,21,27,29\}$ $R_9 = \{2,16,17,25,28,30\}$
$R_{10} = \{3,4,27,29\}$ $R_{11} = \{11,18,27,30\}$ $R_{12} = \{8,12,13,14,16,20,23\}$
$R_{13} = \{1,5,6,13,15,19,24\}$ $R_{14} = \{4,7,14,15,19\}$ $R_{15} = \{1,7,10,12,15,16,25\}$

We should distribute the cost C_j among nonzeros of the column j according to a predetermined strategy for example $d_{ij} = \frac{c_j}{|H_j|}$ $i \in R_j$, $j = 1,2,\dots,n$ which means the costs are distributed equally among the nonzeros of each column. Based on this cost allocation strategy the costs allocated to five nonzeros in column 1 are $\frac{8}{5}$ likewise the costs allocated to each nonzero in all 30 columns are presented as the following row vector:

$\left[\frac{8}{5},\frac{8}{3},\frac{8}{2},\frac{8}{4},\frac{8}{3},\frac{8}{2},\frac{8}{3},\frac{8}{2},\frac{8}{3},\frac{8}{4},\frac{8}{1},\frac{8}{4},\frac{10}{4},\frac{10}{3},\frac{10}{4},\frac{10}{6},\frac{10}{3},\frac{10}{3},\frac{10}{3},\frac{10}{5},\frac{10}{2},\frac{10}{4},\frac{10}{3},\frac{10}{3},\frac{12}{4},\frac{12}{4},\frac{12}{3},\frac{12}{3},\frac{12}{4},\frac{12}{5}\right]$

Therefore,

$d_{11} = \frac{8}{5}, d_{15} = \frac{8}{3}, d_{110} = \frac{8}{4}, d_{112} = \frac{8}{3}, d_{114} = \frac{10}{3}, d_{116} = \frac{10}{6}, d_{118} = \frac{10}{3}, d_{120} = \frac{10}{5}, d_{122} = \frac{10}{4}, d_{125} = \frac{12}{4}$, $d_{130} = \frac{12}{5}$ and $D_1 = min\left\{\frac{8}{5},\frac{8}{3},\frac{8}{4},\frac{8}{3},\frac{10}{3},\frac{10}{6},\frac{10}{3},\frac{10}{5},\frac{10}{4},\frac{12}{4},\frac{12}{5}\right\} = \frac{8}{5}$ and the values of D's corresponding to other rows are

$D_2 = \frac{10}{6}, D_3 = \frac{10}{5}, D_4 = \frac{8}{5}, D_5 = \frac{10}{4}, D_6 = \frac{8}{4}, D_7 = \frac{10}{6}, D_8 = \frac{8}{5}, D_9 = \frac{10}{6}, D_{10} = \frac{8}{4}, D_{11} = \frac{12}{5}, D_{12} = \frac{10}{6}, D_{13} = \frac{8}{5}, D_{14} = \frac{8}{4}$, $D_{15} = \frac{8}{5}$

Therefore the lower bound can be computed as follows:

$$\frac{8}{5}+\frac{10}{6}+\frac{10}{5}+\frac{8}{5}+\frac{10}{4}+\frac{8}{4}+\frac{10}{6}+\frac{8}{5}+\frac{10}{6}+\frac{8}{4}+\frac{12}{5}+\frac{10}{6}+\frac{8}{5}+\frac{8}{4}+\frac{8}{5} = 27.56$$

The optimal objective function value is 42.

5. Two strategies for cost allocation

Although there are infinite number of ways to allocate cost C_j among nonzero entries of column j, two typical strategies are described below:

Strategy 1. In this strategy the cost C_j is distributed equally between $|H_j|$ nonzeros of column, j that is $d_{ij} = \frac{c_j}{|H_j|} \quad j = 1, \ldots., n \;,\; i \in H_j$.

Network relaxations of the set problems do not produce good lower bounds when using this strategy whereas the lower bound obtained here is much stronger.

Strategy 2. Let $R'_i = \left\{ j \mid a_{ij} = 1, j = 1,2, \cdots, [\frac{n}{3}] \right\} \quad i = 1,2, \cdots, m$ where [x] is the greatest integer less than or equal to x. In this strategy the cost allocated to i^{th} nonzero in column j is

$$d_{ij} = \frac{\frac{1}{(\log(1.5+|R'_i|))^{4/9}}}{\sum_{k \in H_j} \frac{1}{(\log(1.5+|R'_k|))^{4/9}}} \quad j = 1, \ldots, n, \text{ and } i \in R_j$$

The rationale behind the above choice of mathematical function and figures are explained in[14] and |X| is the cardinality of set X.

6. Computational experiment

The proposed relaxation was tested on a number benchmark instances taken from Powers [13], Paixao [12], Balas and Ho [1], and Beasley [4]. DUTY problems are two part duty crew scheduling problems randomly generated by Djannaty [9].

Table 1. Comparing proposed relaxation with shortest route relaxation

Prob. Name	Size of Problems.	No. of Nonzero	Strat. 1 Djanaty	St. 1 of Proposd	St. 10 of Djanaty	St. 10 of Proposd	Opt.IP value
AIR01	159×416	2203	12188.7	12188	12830.0	12131	16610
RDM3	101×109	784	35.68	35.66	66.04	70.93	96
RDM4	100× 106	742	38.72	37.89	61.11	62.79	97
RDM6	100× 106	884	47.26	46.48	64.61	65.39	99
RDM7	98×98	704	38.72	35.40	56.40	57.89	87
SCP51	200×2000	11955	113.62	110.62	129.05	134.95	253
SCPA1	300×3000	18000	97.72	97.31	105.2	106.45	253
SCPB1	300×3000	47921	22.17	22.00	22.97	22.82	69
SCPE1	50×500	5414	2.97	2.95	2.95	2.86	5
DUTY3	200×2000	17700	159.4	101.46	201.11	144.9	260
DUTY5	300×2000	17896	270.9	182.72	388.28	311.22	523

Columns 1,2,3, and 8 show characteristics of the test problems, columns 4 and 5 Represent lower bounds obtained by the shortest route relaxation and the proposed Relaxation, respectively, using the same strategy. Except for Duty problems lower Bounds are almost the same. This reveals the fact that the two relaxations are Taking advantage of segments of length one in finding lower bounds and the difference in the last two rows is because these problems have no segments of length one in any column. Columns 7 and 8 present the best strategies of the shortest route relaxation and the proposed relaxation [9]. The above computational experiment proves that lower bounds similar to the shortest route relaxation can be achieved by our proposed relaxation without time consuming computations such as column decomposition, producing the network, storage and retrieval of the network, and finding the shortest route. In addition, less memory usage is another advantage of the proposed relaxation.

7. Conclusions

Network based relaxations attract attention because of their speed, however the proposed relaxation is much faster for not using column decomposition, network optimization, and network algorithms. The proposed relaxation is very flexible and can replace the shortest route relaxation to find the solution of the set problems.

It can also be used to obtain upper bounds for the set packing problem. It is proposed that our relaxation be utilized in a tree search algorithm to find the exact solution of the set problems. It was demonstrated that the proposed relaxation can produce similar bounds to those of network relaxations.

Conflict of Interests

The authors declare that there is no conflict of interests.

REFERENCES

[1] Balas, E. and Ho, A.: Set covering algorithms, using cutting planes, heuristics and subgradient optimization: A new computational study, Mathematical Programming Study, 12 (1980), 37-60.

[2] Balas, E. and Padberg, W.: set partitioning, A survey. SIAM Review, 18(1976), 710-761.

[3] Beasley. J. E.: An algorithm for set covering problems, European Journal of Operational Research, 31 (1987), 85-93.

[4] Beasley. J. E. and Chu, P.C.: A genetic algorithm for the set covering problem, European Journal of Operational Research, 94(1996), 392-404.

[5] orneman. J, Chrobak. M, Vedova.G.D, Figueroa, A. and Jiang: Probe selection algorithms with applications in the analysis of microbial communities, 17(2001), 39-48.

[6] Boros. E., Hammer. P.L., Ibaraki. T. and Kogan, A.: Logical Analysis of Numerical Data, Mathematical Programming, 79(1997), 163-190.

[7] Ceria. S, Nobili. P and Sassano. A.: Annotated bibliographies in Combinatorial Optimization, (1997), 415-428.

[8] Darby-Dowman and Mitra Gautum.: An Extension of Set Partitioning with Application to Scheduling Problems. European Journal of Operational Research, 21(1985), 200-205.

[9] Djannaty, F.: Network based heuristics for the set covering problem, Phd thesis, Brunel University, (1997).

[10] El-Darzi. E. and Mitra. G.: Graph Theoretic Relaxations of Set Covering and Set covering and set partitioning problem, European Journal of Operational Research, 87(1995), 109-121.

[11] Nemhausewer,G.L, and Wolsey L.A. (1988), Integer and combinatorial optimization, Wiley, 1988.

[12] Paixao. J.: algorithm for large scale set covering problems, Imperial College,London, internal report, (1983).

[13] Powers. D.: Investigation and construction of set covering and set partitioning test problems, BSc dissertation, Brunel University, (1987).

[14] Yarali, M.: Strong Cost Allocation Strategies of the Assignment Network Relaxation of the Set Covering Problem, M.s.c thesis, Payame Noor University, (2012).