Benders Decomposition Approach to Set Covering Problems Satisfying Almost the Consecutive Ones Property*

Salim Haddadi and Nacira Hamidane
University of the 8^{th} of May, 1945
Department of Computer Science
Guelma, Algeria
salim.haddadi@yahoo.com

March 22, 2010

Abstract

Although in the actual phase, we are evaluating it, the algorithm presented in this paper is designed for solving real world set covering instances arising from railway public transportation in Germany (Deutsche Bahn). These instances, with large sizes, are characterized by a special pattern. The binary matrices they exhibit, have "almost" consecutive ones property. Taking advantage of this nice structure, we present a decomposition of the SCP into a mixed linear-integer program, and we present a Benders-like algorithm. We end the paper with the computational experience where twenty randomly generated instances are run.
Key Words Set Covering, Benders Decomposition, Consecutive Ones Property, Consecutive Block Minimisation..

1 Introduction

Let A be a binary $m \times n$-matrix and, without loss of generality, let $c_j > 0, j = 1, \cdots, n$, be a cost associated with column j. The set covering problem (SCP) is to cover the rows of A by a subset of the columns at minimum cost. The mathematical model is:

$$\max \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \qquad i = 1, \cdots, m \tag{1}$$

$$x_j \in \{0, 1\} \qquad j = 1, \cdots, n, \tag{2}$$

where the n variables are defined by

*This paper is part of a doctoral dissertation prepared by the second author.

$$x_j = \begin{cases} 1 & \text{if column } j \text{ is in the cover} \\ 0 & \text{otherwise.} \end{cases}$$

The SCP is well known to be hard from theoretical, as well as practical, point of view. It has applications, which are surveyed in [3], in a huge number of domains such as crew scheduling, location of emergency facilities, assembly line balancing, information retrieval, political districting, simplification of boolean expressions, vehicle routing, steel production, traffic assignment in satellite communication systems, and so on, see also the recent papers [1,4 and 9]. Most of the significative algorithms and heuristics are surveyed in [2]. Since then, we noticed that no exact algorithm has been published, and we have found two recent evolutionary heuristics [7,8].

Our goal in this paper, is to present an exact algorithm capable of solving the real world set covering instances arising from railway public transportation in Germany (Deutsche Bahn). These instances, though having a large size, are characterized by a special pattern. The binary matrices they exhibit, have, as was observed in [8], "almost" consecutive ones property (COP). In section 2, we enlighten this fact, and show how can we decrease the number of blocks of consecutive ones. We present, in section 3, the decomposition of the SCP into a mixed linear-integer program, and we present the Benders algorithm. In section 4, we end the paper with the computational experience.

2 Binary matrices having almost the COP

Consider a binary $m \times n$-matrix A. Without loss of generality we may assume that A has no zero rows, nor does it have a zero column. A block of consecutive ones in row i (bco for short) of the binary matrix A is a maximal sequence of ones occurring consecutively. Formally, it is any sequence of entries $a_{ip}, a_{i,p+1}, \cdots, a_{iq}$ in row i satisfying the following:

(i) $a_{ij} = 1, p \leq j \leq q$

(ii) either $p = 1$ or $a_{i,p-1} = 0$

(iii) either $q = n$ or $a_{i,q+1} = 0$.

The matrix A is said to have the consecutive ones property (COP for short) if there exists a permutation of the columns of A so that the ones occur consecutively in every row (in other words, the matrix A has m bco's). Using PQ-trees we can easily (in time linear in the density of A) recognize binary matrices having the COP. Furthermore, the recognition algorithm provides the permutation which leaves the ones appear consecutively in every row. Therefore, without loss of generality, we may assume that a binary matrix having the COP has the property that the ones already occur consecutively in every row.

Most of the binary matrices do not have the COP, and we are interested in the problem of finding a permutation of the n columns of A which minimizes the number of bco's, a well known problem called 'Consecutive Block Minimization (CBM)' which is NP-hard. In [6] we polynomially transformed CBM to the traveling salesman problem satisfying the triangle inequality. Since this polynomial transformation preserves the ratios of approximation, it constitutes a 1.5-approximation for CBM.

The concept of binary matrices having almost the COP appears for the fist time in [8] in the context of SCP's arising from railway transportation problems. Binary matrices obtained in this way exhibit a special shape. Most of the rows have exactly one bco, and each of the remaining rows has a few number of bco's, far less than n. Binary matrices of this kind are said to have almost the COP. Not surprisingly, the SCP with instances having almost the COP is NP-hard (to see this, think of the vertex cover which is a SCP on matrices having two 1's per row, i.e. every row has either one or two bco's).

3 Benders Decomposition

Without loss of generality, suppose that each of the first p rows of A has one bco, and each of the remaining $m-p$ rows has more than one bco ($m-p \ll m$). In every one of the last $m-q$ rows where there are more than one bco, we define a hole to be any maximal sequence of consecutive 0's between two bco's (a hole must exist since there are more than one bco). Our idea is to fill every hole by inserting 1's, so that the resulting matrix has the COP, and attach to each of the block of ones inserted a new binary coupling variable.

Suppose there are h_i holes in row i of the matrix $A, i = p+1, \cdots, m$. Counting the number of bco's, we find (since there are no zero rows)

$$p + \left(m - p + \sum_{i=p+1}^{m} h_i \right) = m + \sum_{i=p+1}^{m} h_i.$$

The total number of holes in A is $h = \sum_{i=p+1}^{m} h_i$. We shall identify every hole by a subset of the set of the columns $J = \{1, \cdots, n\}$. So, let the k^{th} hole in row i be $H_i^k \subset J, (k = 1, \cdots, h_i), (i = p+1, \cdots, m)$. We transform the SCP into the following mixed binary integer program (call it P):

$$\begin{array}{rll}
\min \sum_{j=1}^{n} c_j x_j & & \\
\sum_{j=1}^{n} a_{ij} x_j & \geq 1 & i = 1, \cdots, p \\
\sum_{j=1}^{n} a_{ij} x_j + \sum_{k=1}^{h_i} \sum_{j \in H_i^k} x_j - \sum_{k=1}^{h_i} y_i^k & \geq 1 & i = p+1, \cdots, m \\
\sum_{j \in H_i^k} x_j - y_i^k & \leq 0 & k = 1, \cdots, h_i, i = p+1, \cdots, m \\
x_j & \in \{0,1\} & j = 1, \cdots, n \\
y_i^k & \in \mathbb{Z}_+ & k = 1, \cdots, h_i, i = p+1, \cdots, m
\end{array}$$

There are as many binary variables y_i^k as holes in the binary matrix of the SCP. In condensed form, problem P reads

$$
\begin{array}{rcll}
\min cx & & & \\
A_0x & \geq & e_1 & \\
A_1x - B_1y & \geq & e_2 & (3)\\
A_2x - I_hy & \leq & 0 & (4)\\
x & \in & \{0,1\}^n & \\
y & \in & \mathbb{Z}_+^h &
\end{array}
$$

where e_1, e_2 are respectively the p(resp. $m-p$)-column vector of 1's, and I_h is the $h \times h$ identity matrix. The sizes of the matrices A_0, A_1, A_2, B_1 are respectively $p \times n, (m-p) \times n, h \times n, (m-p) \times h$. Constraints (3) and (4) together replace the original $m-p$ last constraints of the SCP. To see this subtract (4) from (3). From this observation, it is easy to see that problems SCP and P are equivalent. Since the block

$$
\begin{pmatrix} A_0 \\ A_1 \\ A_2 \end{pmatrix}
$$

has the COP, it is totally unimodular and thus we can rewrite problem P, which is a mixed integer linear program as follows

$$
\begin{array}{rcl}
\min cx & & \\
A_0x & \geq & e_1 \\
A_1x - B_1y & \geq & e_2 \\
A_2x - I_hy & \leq & 0 \\
x & \geq & 0 \\
y & \in & \mathbb{Z}_+^h.
\end{array}
$$

Before going any further, let us present an example. Consider the following instance of the SCP

$$
\begin{array}{lccccccccl}
\min & 5x_1 & +3x_2 & +2x_3 & +x_4 & +x_5 & +2x_6 & +3x_7 & & \\
 & & x_2 & +x_3 & +x_4 & +x_5 & & & & \geq 1 \\
 & x_1 & +x_2 & +x_3 & +x_4 & & & & & \geq 1 \\
 & & & x_3 & +x_4 & +x_5 & +x_6 & +x_7 & & \geq 1 \\
 & & x_2 & +x_3 & +x_4 & +x_5 & +x_6 & & & \geq 1 \\
 & x_1 & & +x_3 & +x_4 & & & +x_7 & & \geq 1 \\
 & & x_2 & +x_3 & & & +x_6 & & & \geq 1 \\
 & x_1, & x_2, & x_3, & x_4, & x_5, & x_6, & x_7 & & \in \{0,1\}.
\end{array}
$$

Here $m = 6, n = 7$. The binary constraints matrix has $p = 4$ first rows containing one bco each. The remaining $m - p = 2$ rows have $h = 3$ holes, two in the fifth row and one in the sixth, and we have $H_5^1 = \{2\}, H_5^2 = \{5,6\}, H_6^1 = \{4,5\}$. The proposed decomposition (filled holes are boldfaced) results in the problem

P shown below

$$
\begin{array}{lccccccccccl}
\min & 5x_1 & +3x_2 & +2x_3 & +x_4 & +x_5 & +2x_6 & +3x_7 & & & & \\
 & & x_2 & +x_3 & +x_4 & +x_5 & & & & & & \geq 1 \\
 & x_1 & +x_2 & +x_3 & +x_4 & & & & & & & \geq 1 \\
 & & & x_3 & +x_4 & +x_5 & +x_6 & +x_7 & & & & \geq 1 \\
 & & x_2 & +x_3 & +x_4 & +x_5 & +x_6 & & & & & \geq 1 \\
 & x_1 & +\mathbf{x}_2 & +x_3 & +x_4 & +\mathbf{x}_5 & +\mathbf{x}_6 & +x_7 & -y_5^1 & -y_5^2 & & \geq 1 \\
 & & x_2 & +x_3 & +\mathbf{x}_4 & +\mathbf{x}_5 & +x_6 & & & & -y_6^1 & \geq 1 \\
 & & x_2 & & & & & & -y_5^1 & & & \leq 0 \\
 & & & & & x_5 & +x_6 & & & -y_5^2 & & \leq 0 \\
 & & & & x_4 & +x_5 & & & & & -y_6^1 & \leq 0 \\
 & x_1, & x_2, & x_3, & x_4, & x_5, & x_6, & x_7 & & & & \geq 0 \\
 & & & & & & & & y_5^1, & y_5^2, & y_6^1 & \in \mathbb{Z}_+
\end{array}
$$

where

$$
A_0 = \begin{pmatrix} & 1 & 1 & 1 & 1 & & \\ 1 & 1 & 1 & 1 & & & \\ & & 1 & 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 & 1 & 1 & \end{pmatrix}
$$

$$
A_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 & 1 & 1 & \end{pmatrix} \quad B_1 = \begin{pmatrix} 1 & 1 & \\ & & 1 \end{pmatrix}
$$

$$
A_2 = \begin{pmatrix} & 1 & & & & & \\ & & & & 1 & 1 & \\ & & & 1 & 1 & & \end{pmatrix} \quad I_3 = \begin{pmatrix} 1 & & \\ & 1 & \\ & & 1 \end{pmatrix}
$$

For the convenience of presenting the algorithm, we shall rewrite problem P as follows

$$
\begin{array}{rcl}
\min cx & & \\
Ax - By & \geq & e \\
A_2 x - I_h y & \leq & 0 \\
x & \geq & 0 \\
y & \in & \mathbb{Z}_+^h .
\end{array}
$$

where

$$
A = \begin{pmatrix} A_0 \\ A_1 \end{pmatrix} \text{ and } B = \begin{pmatrix} \mathbf{0} \\ B_1 \end{pmatrix}
$$

where $\mathbf{0}$ is the $p \times h$-matrix whose elements are 0's. Benders decomposition paradigm is well known, so we merely present the algorithm steps.

Observe that problem (X) has always a feasible solution $u_1 = 0, u_2 = 0$ (since $c > 0$). Therefore, either does it have an optimal solution, or is it unbounded. On the other hand, assuming no zero rows in the binary constraints matrix, the SCP has an optimal cover. Consequently, problem (Y) must have an optimal solution at any iteration, and the only criteria to finish is to attain the same objective function value for the two problems.

4 Computational experience

We coded our algorithm in C on HP Compaq Core Duo 1.6GHz and run it on a set of twenty randomly generated instances. The numbers m, n, p were fixed

Input: positive integers m, n, matrix constraint of the SCP and cost vector c
Output: optimal cover $\bar{x}$ and its cost $\bar{c}$

- try to reduce the size of the set covering problem by using logical tests;
- try to decrease the number of holes by approximating the underlying CBM problem;
- Use the greedy heuristic to find an approximate cover $\tilde{x}$;
- $\bar{y}^{(0)} \leftarrow A_2\tilde{x}$ and $s \leftarrow 0$;
- repeat

  solve problem

$$(X)\begin{cases} \max\left(e + B\bar{y}^{(s)}\right)u_1 - A_2u_2 \\ A^T u_1 - A_2^T u_2 \leq c \\ u_1, u_2 \geq 0 \end{cases}$$

  if (X) has an optimal solution $\left(\bar{u}_1^{(s+1)}, \bar{u}_2^{(s+1)}\right)$ with an objective function value $\bar{c}$ then add to the master problem

$$(Y)\begin{cases} \min z \\ (\bar{u}_1^{(k)}B - \bar{u}_1^{(k)}A_2)y - z \leq -e\bar{u}_1^{(k)} k = 1, \cdots, s \\ z \geq 0 \\ y \in \mathbb{Z}_+^h \end{cases}$$

  the cut $(\bar{u}_1^{(s+1)}B - \bar{u}_2^{(s+1)}A_2)y - z \leq -e\bar{u}_1^{(s+1)}$;

  if (X) is unbounded, let $\left(\bar{u}_1^{(s+1)}, \bar{u}_2^{(s+1)}\right)$ be the actual extreme point and $\left(\bar{u}_1^{(s+2)}, \bar{u}_2^{(s+2)}\right)$ be the extreme ray, then add to (Y) the two cuts $(\bar{u}_1^{(s+1)}B - \bar{u}_2^{(s+1)}A_2)y - z \leq -e\bar{u}_1^{(s+1)}$and $(\bar{u}_1^{(s+2)}B - \bar{u}_2^{(s+2)}A_2)y \leq -e\bar{u}_1^{(s+2)}$;

  Solve problem (Y) and let $\bar{y}^{(s+1)}$be an optimal solution with objective function value $\bar{z}$;

  $s \leftarrow s + 1$;
- until $\left(\bar{c} = \bar{z}\right)$:

Dens.	Number of holes before heuristic	Number of holes after heuristic	Number of cuts	Time
5%	41	26	13	51.23
	35	25	10	42.36
	41	29	28	80.03
	34	28	10	35.23
	38	27	14	55.76
Mean values			15.0	52.9
10%	36	28	10	52.30
	41	32	17	71.26
	50	33	23	87.33
	68	44	22	75.16
	40	30	17	70.33
Mean values			17.8	71.2
15%	69	44	17	84.66
	49	38	18	82.30
	52	36	22	90.63
	62	45	28	96.56
	80	55	21	90.33
Mean values			21.2	88.9
20%	128	84	19	151.83
	87	69	24	122.26
	115	80	32	163.26
	96	63	26	118.80
	101	65	29	158.70
Mean values			26	143.0

Table 1: Computational results on twenty randomly generated instances

once for all to 50, 200, 45 respectively. First, the binary matrices generated have the COP, then ∂% of the ones in the five last rows are replaced by 0's ($\partial = 5, 10, 15, 20$). The binary matrices are preprocessed in order to decrease the crucial number of holes, by transforming the underlying CBM problem to the traveling salesman problem, and applying a simple greedy heuristic to find an approximate tour. During the execution of our algorithm, the subprograms (related to vector x) are computed using a rudimentary simplex algorithm (method of tableaus) and the master programs are resolved with a rudimentary implicit enumeration method. This explains why executing times are rather large, the instances being of small or medium size. What is interesting is the small number of cuts needed to confirm the optimality.

Our next task is to test this algorithm (after fine-tuning the preprocessing phase using logical tests to reduce the size of the problem, and using Cplex instead of the rudimentary methods used above) on the real-world instances obtained from Ruf and Schöbel (see [8]).

References

[1] Blazsik, Z. and B. Imreh (1996) A Note on Connection Between Process Network Synthesis and Set Covering Problems, Acta Cybernetica 12, 309-312.

[2] Caprara, A., M. Fischetti and P. Toth (2000) Algorithms for the Set Covering Problem, Annals of Operations Research 98, 353-371.

[3] Ceria, S., P. Nobili and A. Sassano (1997) Set Covering Problem. In M. Dell'Amico, F. Mafioli and P. Toth (eds.) Annotated Bibliographies in Combinatorial Optimization, John Wiley and Sons, New York, 415-428.

[4] Chen, L. and J. Crampton (2009) Set Covering Problems in Role-Based Access Control, In Proceedings of the 14th European Symposium on Research in Computer Security (ESORICS09), 689-704.

[5] Gouwanda, D. and S.G. Ponnambalam (2008) Evolutionnary Search Techniques to Solve Set Covering Problems, World Academy of Science, Engineering and Technology 39, 20-25.

[6] Haddadi, S. (2008) Consecutive Block Minimization is 1.5-approximable, Information Processing Letters 108, 132-135.

[7] Lan, G., G.W. DePuy and G.E. Whitehouse (2007) An Effective and Simple Heuristic for the Set Covering Problem, European Journal of Operational Research 176, 1387-1403.

[8] Ruf, N. and A. Schöbel (2004) Set Covering with almost Consecutive Ones Property, Discrete Optimization 15, 215-228.

[9] Yang, J. and J.Y-T. Leung (2003) A Generalization of the Weighted Set Covering Problem, Wiley Periodicals (Draft).