Report of assignment 4: Benders' decomposition for Cell Suppression Problem

Luis Rojo-González[a,b]

[a] *Department of Industrial Engineering, Universidad de Santiago de Chile, Chile*
[b] *Department of Statistics and Operational Research, Universitat Politècnica de Catalunya, Spain*

Abstract

The Cell Suppression Problem (CSP) is a very large Mixed-Integer Linear Problem arising in statistical disclosure control. However, CSP has the typical structure that allows application of the Benders' decomposition, which is known to suffer from oscillation and slow convergence, compounded with the fact that the master problem is combinatorial. We use the relaxation of the lower bounds of protection levels to generate a Sub Problem to generate cuts to help the solver to get fast with 50 iterations tops. Experiments were developed on three instances with different setting and size. Obtained results show non-convergence at the end of the cycle but providing a feasible solution which does not change even before the last iteration.

1. Introduction

The Cell Suppression Problem (CSP) [6, 4, 5, 3] is a statistical disclosure method based on removing the minimum amount of information (measured as a function of the number of cells or cell values) that makes the resulting table safe. CSP has relatively few (but still in the thousands) binary variables, but very many (millions) continuous ones. Furthermore, once the binary variables are fixed, the problem in the continuous ones decomposes into many independent subproblems [1]. For this reason, we apply Benders' decomposition to get feasible cuts that provide facets of the convex space to get fast the procedure of the solver.

2. Statement of the problem

Let a data array of points $\mathcal{N}$, $X = \{x_1, x_2, \dots, x_{\mathcal{N}}\}$ of length $|\mathcal{N}|$, where every point $x_{i=\{1,\dots,N\}}$ has to be protected within an interval (l_i, u_i). Also, be a subset $\mathcal{S} \subset \mathcal{N}$ of points we must protect with respectively level protection (lpl_s, upl_s), $\forall s \in \mathcal{S}$, called sensitive points. Each one of these points have a protecting cost c_i and a final protection level $x_i \in \mathbb{R}_+$. More formally, protection level is within a lower bound x_i^l as well as an upper bound x_i^u which can be interpreted as margins.

3. Linear programming (LP) model

The goal is to protect the least amount of points considering those that are necessary to protect. So, consider a binary variable y_i that takes the value of 1 if point $i \in \mathcal{N}$ is protected, 0 otherwise. Be $x_i^l, x_i^u \in \mathbb{R}_+$, $\forall i \in \mathcal{N}$ protecting levels of each point i. Therefore, we have the following mathematical model (in their matrix form)

$$min_{y,x^l,x^u} \qquad \sum_{i \in \mathcal{N}} c_i y_i \tag{1}$$

Email address: `luis.rojo.g@usach.cl` (Luis Rojo-González)

$$
\begin{aligned}
s.t.(\forall s \in \mathcal{S}) \quad & Ax^{l,s} = 0 && && (2)\\
& (l_i - a_i)y_i \leq x_i^{l,s} \leq (u_i - a_i)y_i && \forall i \in \mathcal{N} && (3)\\
& x_s^{l,s} \leq -lpl_s && && (4)\\
& Ax^{u,s} = 0 && && (5)\\
& (l_i - a_i)y_i \leq x_i^{u,s} \leq (u_i - a_i)y_i && \forall i \in \mathcal{N} && (6)\\
& x_s^{u,s} \leq -upl_s && && (7)\\
& y_i \in \{0,1\} && \forall i \in \mathcal{N} && (8)
\end{aligned}
$$

This model guarantee that each protected point has a protecting level within corresponding bounds such that the aim is to protect the least amount of points.

4. Benders' decomposition

Benders' decomposition [2] is applied to those mathematical models which consider non-negative variables as well as integer variables, such as binary variables. It is understood as a kind of relaxation where a subproblem is generated considering a part of the original problem and solving this problem to generate and to add cuts in the original problem (known as master problem).

On this case, we consider the above LP to generate a relaxation such that we relax those variables with an superscript l such that for all $s \in \mathcal{S}$ the subproblem is

$$
\begin{aligned}
-lpl_s \geq min_x \quad & x_s && && (9)\\
s.t. \quad & Ax = 0 && && (10)\\
& (l_i - a_i)y_i \leq x_i \leq (u_i - a_i)y_i && \forall i \in \mathcal{N} && (11)\\
& x_s \leq -lpl_s && && (12)\\
& y_i \in \{0,1\} && \forall i \in \mathcal{N} && (13)
\end{aligned}
$$

and its associated dual problem is

$$
\begin{aligned}
max_{\lambda,\mu} \quad & \sum_{i \in \mathcal{N}} \left((l_i - a_i)\mu_i^l - (u_i - a_i)\mu_i^u)y_i\right) && && (14)\\
s.t. \quad & \sum_j A_{j,i}\lambda_j + \mu_s^l + \mu_s^u = 1 && \forall s \in \mathcal{S} && (15)\\
& \sum_j A_{j,i}\lambda_j + \mu_i^l + \mu_i^u = 0 && \forall i \in \mathcal{I} \setminus \mathcal{S} && (16)\\
& \mu_i^l, \mu_i^u \geq \forall i \in \mathcal{I} && && (17)
\end{aligned}
$$

where equation (18) provides a valid cut to the problem

$$
-lpl_s \geq \sum_{i \in \mathcal{N}} \left((l_i - a_i)\mu_i^l - (u_i - a_i)\mu_i^u)y_i\right), \quad \forall s \in \mathcal{S} \tag{18}
$$

Also note that if we use the relaxation of those variables with superscript u instead of l the cut would be provided by equation (19)

$$
upl_s \leq \sum_{i \in \mathcal{N}} \left(-(l_i - a_i)\mu_i^l + (u_i - a_i)\mu_i^u)y_i\right), \quad \forall s \in \mathcal{S} \tag{19}
$$

5. Computational experiments

Three computational experiments were carried out considering instances of different setting and size. The implementation of the mathematical model was in AMPL version 20180110 and was solved using solver CPLEX 12.8.0.0 for integer programming. It was executed in a Macbook Air Intel Core i5, 1.6 Ghz, 8GB RAM.

6. Results

Obtained results, after 50 iterations on each instance give 9 points protected in Small instance, 6 points protected in Example2D instance and 15 protected points in Targus instance. Also, Figure 1 show the evolution of bounds on each iteration for Small instance, Example2D instance and Targus instance, respectively.

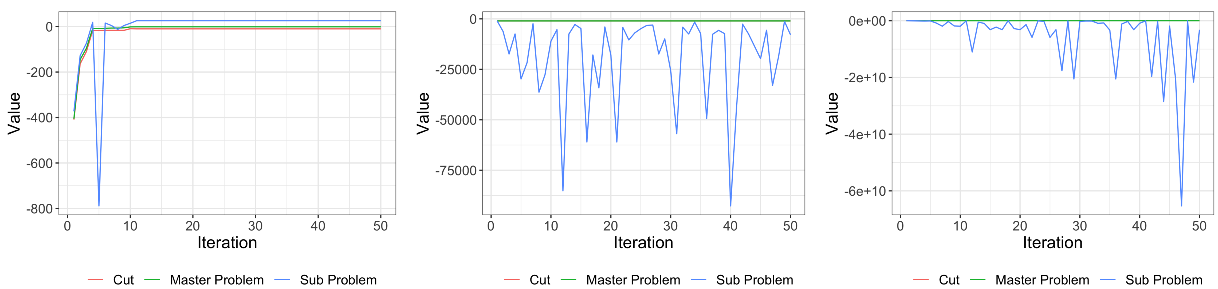

Figure 1: Evolution of each bound throughout iterations.

In Small instance, $s = \{1, 5, 9, 24\}$ and protected points are $i = \{1, 3, 5, 9, 10, 23, 24, 25, 26\}$. In Example2D instance, $s = \{16, 22, 27, 30\}$ and protected points are $i = \{3, 16, 22, 27, 28, 30\}$. In targus instance, sensitive points are $s = \{19, 20, 21, 24, 25, 27, 38, 39, 42, 45, 51, 52, 133\}$ where protected points are $i = \{19, 20, 21, 24, 25, 27, 38, 39, 42, 45, 51, 52, 99, 124, 133\}$.

7. Conclusion

The CSP mathematical model is a big model which involves several integer variables as well as non-restricted variables. For that, any solver should consume a lot of memory to solve the instance. Nonetheless, Benders' decomposition provides feasible cuts to find the convex space and therefore the optimal solution. Thus, solving a small model we can find a (feasible) solution in few iterations.

References

[1] D. Baena, J. Castro, and A. Frangioni. Stabilized Benders methods for large-scale combinatorial optimization, with application to data privacy. `http://www-eio.upc.edu/~jcastro/publications/reports/dr2017-03.pdf`, 2017, uptadet 2018. [Online; accessed 5-January-2019].

[2] J. Benders. Partitioning procedures for solving mixed-variables programming problems ‘. 1962.

[3] J. Castro. A shortest-paths heuristic for statistical data protection in positive tables. *INFORMS Journal on Computing*, 19(4):520–533, 2007.

[4] M. Fischetti and J. J. Salazar. Models and algorithms for the 2-dimensional cell suppression problem in statistical disclosure control. *Mathematical Programming*, 84(2), 1999.

[5] M. Fischetti and J. J. Salazar. Solving the cell suppression problem on tabular data with linear constraints. *Management Science*, 47(7):1008–1027, 2001.

[6] J. P. Kelly, B. L. Golden, and A. A. Assad. Cell suppression: Disclosure protection for sensitive tabular data. *Networks*, 22(4):397–417, 1992.