A Fix and Relax Heuristic for Controlled Tabular Adjustment

European Conference on Operational Research, Lithuania 2012

Daniel Baena and Jordi Castro

{daniel.baena@upc.edu // jordi.castro@upc.edu}
Dept. of Statistics and Operations Research
Universitat Politècnica de Catalunya
Barcelona

8-11 July, 2012

Supported by MICINN MTM2009-08747 and SGR-2009-1122 grants.

Contents

Statistical Disclosure Control

- Protect confidential information in released data.
- National Statistical Agencies (NSAs):
  - Need to release a large amount of data.
  - Forced (by law) to guarantee that no confidential information from any respondent is disclosed.
- Two types of data:
  - Disaggregated data (microdata): Contains individual information.
  - Aggregated data (macrodata): **Tabular data** by crossing categorical variables.

Tabular data protection methods

- Why disclosure risk in aggregated data? Example:

2D magnitude table: average salary profession x age

	P_1	P_2	P_3	Total
A_1	···	···	···	···
A_2	···	38.000€	40.000€	···
A_3	···	39.000€	42.000€	···
Total	···	···	···	···

2D frequency table: number of persons profession x age

	P_1	P_2	P_3	Total
A_1	···	···	···	···
A_2	···	20	1 or 2	···
A_3	···	30	35	···
Total	···	···	···	···

  - If two tables are published, any attacker would know the salary of the unique respondent of cell(A_2,P_3) is 40.000 €
  - If there are two respondents, any of them could deduce the other's salary
- Non-perturbative methods: Don't change the original data, instead suppress minimal data.Cell Suppression Problem (CSP).
- Perturbative methods: Provide an alternative table with minimal modifications. Controlled Tabular Adjustment (CTA)

Controlled Tabular Adjustment: Parameters

- Set of cells $a_i, i = 1, \ldots, n$.
- Linear relations $Aa = b$.
- Lower and upper bound for each cell: l_{a_i} and u_{a_i}.
- Cell weights w_i for cost of adjustment of each cell.
- Set $\mathcal{P} = \{i_1, i_2, \ldots, i_p\} \subseteq \{1, \ldots, n\}$ of indices of sensitive cells.
- Lower and upper protection level for each sensitive cell $i \in \mathcal{P}$: lpl_i and upl_i.

Controlled Tabular Adjustment: Purpose

- To find the **closest safe table to the original one.**
- How? To find released values x_i such that:
  - They are the closest ones to a_i.
  - Satisfy the linear relations $Ax = b$.
  - Satisfy the bounds (lower/upper): $l_{a_i} \leq x_i \leq u_{a_i}$.
  - Satisfy the protection levels: either $x_i \geq a_i + upl_i$ **OR** $x_i \leq a_i - lpl_i$.

The optimization problem CTA can be formulated as (using absolute values of deviations):

$$\begin{array}{rl} Z = \min\limits_{x} & w|x-a| \\ \text{s. to} & Ax = b \\ & l_{a_i} \leq x_i \leq u_{a_i} \quad i = 1, \ldots, n \\ & x_i \leq a_i - lpl_i \textbf{ OR } x_i \geq a_i + upl_i \quad i \in \mathcal{P}. \end{array}$$

Controlled Tabular Adjustment: The MILP model

- Let $z_i = x_i - a_i$. Consider positive and negative deviations $z_i^+, z_i^- : z_i = z_i^+ - z_i^-$ for $i \in N$,
- Introducing binary variables $y_i, i \in \mathcal{P}$ for sensitive cells:
  - $y_i = 1$ the protection sense is "upper": $upl_i \le z_i^+$ and $z_i^- = 0$;
  - $y_i = 0$ the protection sense is "lower": $lpl_i \le z_i^-$ and $z_i^+ = 0$;

The MILP model is:

$$
\begin{array}{rl}
Z = \displaystyle\min_{z^+, z^-, y} & \displaystyle\sum_{i=1}^{n} w_i (z_i^+ + z_i^-) \\
\text{s. to} & A(z^+ - z^-) = 0 \\
& 0 \le z_i^+ \le \quad u_{z_i} \quad i \notin \mathcal{P} \\
& 0 \le z_i^- \le -l_{z_i} \quad i \notin \mathcal{P} \\
& upl_i \; y_i \le z_i^+ \le u_{zi} \; y_i \quad i \in \mathcal{P} \\
& lpl_i (1 - y_i) \le z_i^- \le -l_{zi} (1 - y_i) \quad i \in \mathcal{P} \\
& y_i \in \{0, 1\} \; i \in \mathcal{P}.
\end{array}
$$

Fix and Relax Heuristic: State of the art

- Applied to Project Scheduling Problems:
  - L.F. Escudero, J. Salmeron, On a Fix-and-Relax Framework for a Class of Project Scheduling Problems, Annals of Operations Research, 140, 163-188, 2005.
  - C. Dillenberger, L.F. Escudero, A. Wollensak, W. Zhang, On practical resource allocation for production planning and scheduling with period overlapping setups, European Journal of Operational Research 75,275-286,1994.

Fix and Relax: Motivation

- CTA is a MILP challenging even for tables of moderate size. Finding an optimal (or quasi-optimal) solution may requiere many hours of execution.
- Branch and Cut (BC) scheme to solve MILP eventually becomes inefficient (as number of binary/integer variables increase) because of the exponential growth in the number of nodes to explore.
  - It takes much more computing time and frequently fails to give a solution.
- Recently, a Block Coordinate Descent heuristic(BCD) was successfully applied to CTA. However BCD needs an initial feasible solution.

Fix and Relax: Purpose

- Find an upper bound, hopefully of good quality, to CTA problem in reasonable computing time.
- Based on partitioning the set of binary variables into clusters to selectively explore a smaller BC tree.
- FR solves the MILP in a number of steps, each of which involves a subproblem of smaller complexity than the original MILP.
- Lower Bound: Only at first iteration, because it's a relaxation of the original problem.
- Fix and Relax does not need an initial feasible solution.

Fix and Relax: Algorithm

Input: For a given number of clusters $k \geq 0$ (not neccessarily of the same size), decompose the set of binary variables in k non-disjoint sets $V_1, \ldots, V_k$.

Step 1: Set $r = 1$ and solve subproblem (CTA_r) with integrality constraints for only the reduced subset of binary variables (cluster V_1). The rest are relaxed.

- **Step 1.1:** If feasible, $Z_{LB} = Z^*$ and $r = r + 1$. Otherwise, STOP.

Step 2: Fix values of V_{r-1} at their optimal values. With V_r integer and the rest relaxed solve the new subproblem (CTA_r).

- **Step 2.1:** If $r = k$, set $Z_{UB} = Z^*$ and STOP. Problem CTA is feasible.
- **Step 2.2:** Otherwise, if feasible, set $r = r + 1$ and go to Step 2.
- **Step 2.3:** If infeasible, backward step to redefine the partition structure (join V_{r-1} and V_r).

Computational Results: Details

- Fix and Relax was implemented in C++ using the commercial solver Ilog Cplex 12.4 to solve each subproblem.
- Applied to 1H2D instances (Two-dimensional tables with one hierarchical variable).
- Number and type of clusters tested:
  - Random clusters of size 10, 50 or 100. Same size (except one).
  - Taking into account the structure of 1H2D instances. Binary variables of each 2D table in the same cluster. Maybe different size.
- A big amount of instances were run. Here only a subset reported.
- All runs were carried on a Dell PowerEdge 6950 server, four dual core AMD Opteron 8222 3.0 Ghz processors, 64GB of RAM. Without use of parallelism capabilities.

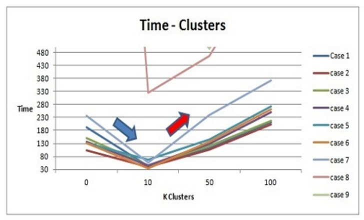
Computational Results: 1H2D symmetric instances

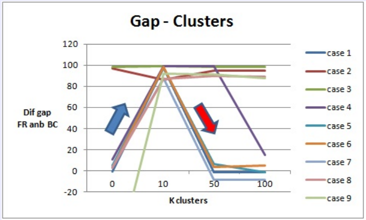
- Random Symmetric ($upl_i = lpl_i$) instances selected.
- Optimality gap of 1%.
- Limit time: 7200 seconds.

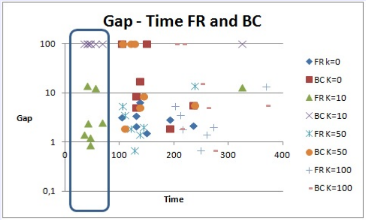
Instance	Cells	Sensitive Cells	Constraints	Non-zero coeffs
Small size instances				
case 1	78761	3840	4504	160064
case 2	77408	3774	4471	157358
case 3	79171	3860	4514	160884
Medium size instances				
case 4	92055	4510	5018	187272
case 5	96696	4737	5109	196554
case 6	97539	4756	4962	197620
Big size instances				
case 7	119238	5842	5551	241638
case 8	130611	12800	5774	264384
case 9	127959	12540	5722	259080

Computational results: 1H2D symmetric instances

- What is, in general, the optimum number of clusters?







- We choose K=10

Computational Results: 1H2D symmetric instances

- Gap is defined as $((UB - LB)/UB)$. Where LB is the best known lower bound computed in our tests.
- Gap of BC computed at the time Fix and Relax found the integer feasible solution.

Instance	Time	gapFR	nodesFR	Ninfeas	gapBC	nodes BC
case1	42.34	2.40	0	0	100.00	960
case2	41.35	13.92	20	0	100.00	699
case3	46.53	1.23	9	0	100.00	854
case4	47.45	0.86	5	0	100.00	100
case5	69.46	2.46	14	0	100.00	501
case6	35.98	1.45	0	0	100.00	1255
case7	57.15	12.57	2	0	100.00	2051
case8	326.32	13.27	51	4	100.00	70
case9	822.21	8.23	394	0	100.00	493

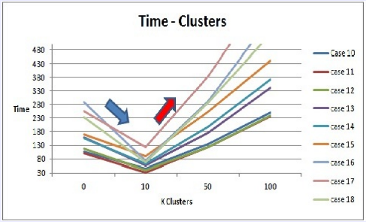
Computational Results: 1H2D asymmetric instances

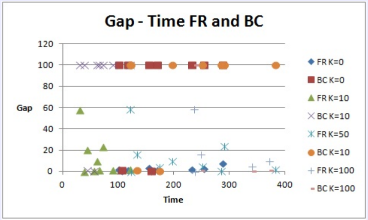
- Random Asymmetric ($upl_i \neq lpl_i$) instances selected.
- Optimality gap of 1%.
- Limit time: 7200 seconds.

Instance	Cells	Sensitive Cells	Constraints	Non-zero coeffs
Small size instances				
case10	73390	3578	4373	149322
case11	72693	3544	4356	147928
case12	74374	3626	4397	151290
Medium size instances				
case13	98226	4812	5139	199614
case14	95166	4662	5079	193494
case15	94911	4650	5074	192984
Big size instances				
case16	133977	6565	5840	271116
case17	131682	6452	5795	266526
case18	126429	6195	5692	256020

Computational results: 1H2D asymmetric instances

- Now, what is the optimum number of clusters?





- Choose K=10, it finds a good feasible solution in a short time

Computational Results: 1H2D asymmetric instances

- Gap is defined as $((UB - LB)/UB)$. Where LB is the best known lower bound computed in our tests.
- Gap of BC computed at the time Fix and Relax found the integer feasible solution.

Instance	Time	gap FR	nodes FR	Ninfeas	gap BC	nodes BC
case 10	45.67	20.73	56	0	1.64	520
case 11	31.45	58.02	0	0	100	0
case 12	39.52	0.22	0	0	100	0
case 13	57.52	0.76	35	0	0.82	2
case 14	62.83	9.58	19	0	100	1136
case 15	91.24	1.61	8	0	100	0
case 16	73.81	23.19	0	2	100	2070
case 17	122.44	2.25	0	0	100	0
case 18	67.51	0.97	0	0	100	0

Conclusions and Extensions

- The field of Statistical Data Protection is a source of real applications of optimization.
- Controlled Tabular Adjustment (CTA) was implemented at UPC for European NSAs and Eurostat.
- Fix and Relax Heuristic: shown to be a succesful heuristic for good solutions to MILP CTA Problem in a reasonable computing time.
- Things to do:
    - What about general tables? Can we expect the same performance?
    - Combine Fix and Relax with other heuristics: Local Branching.
    - Use Fix and Relax solution as warm start for BC or Block Coordinate Descent.

Thanks