Using BCD-CTA for difficult tables:
a practical experiment with a real Eurostat table

Jordi Castro      José A. González      Anco Hundepool

Universitat Politècnica de Catalunya
Barcelona, Catalonia

jordi.castro@upc.edu      jose.a.gonzalez@upc.edu      hundepool@ziggo.nl

This work has been supported by grants MTM2012-31440 of the Spanish research program, and project DwB INFRA-2010-262608 of the EU FP7.
Contents

1. Introduction and motivation
2. Outline of MILP-CTA
3. BCD-CTA
4. Computational results
5. Conclusions and future work
1 Introduction and motivation

2 Outline of MILP-CTA

3 BCD-CTA

4 Computational results

5 Conclusions and future work
Protection of tables published at the EU-level

▶ The goal:

1. At national level: NSIs collect and process the information of each member state (primary and secondary suppressions).
2. The NSIs send their tables to Eurostat.
3. Eurostat compiles the complete European table, with the obligation to guarantee the confidentiality of the national data.

▶ The solution approach of S. Giessing, A.H., J.C. (2009) was to compute rounded figures for the EU-cells at risk.

▶ CTA was used in one of the steps of this solution approach.
The instance that motivated this work


- Three-dimensional table.
  - First: EU-member state (27 plus the EU-total).
  - Second: NACE hierarchical classification (120 codes).
  - Third: size-class (5 codes plus a total)

- Number of cells: $28 \cdot 120 \cdot 6 = 20160$.

- Number of constraints: 8280.

- A.H. found unacceptable long running CTA times for this instance.


- Solution: either use BCD-CTA or try different weights.
1. Introduction and motivation

2. Outline of MILP-CTA

3. BCD-CTA

4. Computational results

5. Conclusions and future work
MILP-CTA: Example

ORIGINAL TABLE. Protection levels: $x_{23} \geq 45$ or $x_{23} \leq 35$

<table>
<thead>
<tr>
<th></th>
<th>$Z_1$</th>
<th>$Z_2$</th>
<th>$Z_3$</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_1$</td>
<td>20</td>
<td>24</td>
<td>28</td>
<td>72</td>
</tr>
<tr>
<td>$E_2$</td>
<td>38</td>
<td>38</td>
<td>40</td>
<td>116</td>
</tr>
<tr>
<td>$E_3$</td>
<td>40</td>
<td>39</td>
<td>42</td>
<td>121</td>
</tr>
<tr>
<td>TOTAL</td>
<td>98</td>
<td>101</td>
<td>110</td>
<td>309</td>
</tr>
</tbody>
</table>
### ORIGINAL TABLE. Protection levels: $x_{23} \geq 45$ or $x_{23} \leq 35$

<table>
<thead>
<tr>
<th></th>
<th>$Z_1$</th>
<th>$Z_2$</th>
<th>$Z_3$</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_1$</td>
<td>20</td>
<td>24</td>
<td>28</td>
<td>72</td>
</tr>
<tr>
<td>$E_2$</td>
<td>38</td>
<td>38</td>
<td>40</td>
<td>116</td>
</tr>
<tr>
<td>$E_3$</td>
<td>40</td>
<td>39</td>
<td>42</td>
<td>121</td>
</tr>
<tr>
<td>TOTAL</td>
<td>98</td>
<td>101</td>
<td>110</td>
<td>309</td>
</tr>
</tbody>
</table>

### PROTECTED TABLE: either ... or ...

<table>
<thead>
<tr>
<th></th>
<th>$Z_1$</th>
<th>$Z_2$</th>
<th>$Z_3$</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_1$</td>
<td>25</td>
<td>24</td>
<td>23</td>
<td>72</td>
</tr>
<tr>
<td>$E_2$</td>
<td>33</td>
<td>38</td>
<td>45</td>
<td>116</td>
</tr>
<tr>
<td>$E_3$</td>
<td>40</td>
<td>39</td>
<td>42</td>
<td>121</td>
</tr>
<tr>
<td>TOTAL</td>
<td>98</td>
<td>101</td>
<td>110</td>
<td>309</td>
</tr>
</tbody>
</table>
Parameters of MILP-CTA

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

Find released values $x$ such that:

- Remain near $a$ for some distance $\ell$.
- Satisfy the linear relations $Ax = b$
- Satisfy the bounds: $l_x \leq x \leq u_x$
- Satisfy the protection levels: either $x_i \geq a_i + upl_i$ or $x_i \leq a_i - lpl_i$.

The optimization problem is:

$$\min_{x} \|x - a\|_{\ell}$$

subject to

$Ax = b$

$l_x \leq x \leq u_x$

$x_i \leq a_i - lpl_i$ or $x_i \geq a_i + upl_i \quad i \in S.$
The MILP-CTA problem for $\ell_1$

- Defining *cell deviations* as: $z = x - a$,
- and introducing binary variables for sensitive cells: $y_i, i \in S$
  
  $y_i = 1 \iff$ *upside* protected: $x_{i} \geq a_{i} + upl_{i}$;
  
  $y_i = 0 \iff$ *downside* protected: $x_{i} \leq a_{i} - lpl_{i}$.

MILP-CTA model:

$$\min_{z^+, z^-, y} \sum_{i=1}^{n} w_i (z_i^+ + z_i^-)$$

subject to

$$A(z^+ - z^-) = 0$$

$$0 \leq z^+ \leq u_z, \quad 0 \leq z^- \leq -l_z$$

$$y \in \{0, 1\}^s$$

$$upl_i y_i \leq z_i^+ \leq u_z i y_i$$

$$lpl_i (1 - y_i) \leq z_i^- \leq -l_z i (1 - y_i) \quad \{i \in S\}$$
Contents

1. Introduction and motivation
2. Outline of MILP-CTA
3. BCD-CTA
4. Computational results
5. Conclusions and future work
CD and BCD algorithms

- **Coordinate descent**: family of optimization algorithms that successively optimize along coordinate directions.
- Very popular in 80s and 90s.
- Recently gained reputation for approximate solutions in big-data.
- **Block coordinate descent (BCD)**: optimize on a block of variables.
- BCD solves a sequence of subproblems on a subset of variables, the other variables kept fixed.
Step 0 Initialization. Set outer iteration counter: $t = 0$. Set initial values, hopefully feasible, to $y$.

Step 1 $t = t + 1$. Set inner iteration counter $i = 0$. Divide $y$ into $k$ blocks: $y = \{y^{1,i}, \ldots , y^{k,i}\}$, not necessarily of the same size.

Step 1.1 $i := i + 1$. Solve CTA with respect to block $y^{i,i}$, taking into account that $y^{j,i}$ is fixed for $j \neq i$.

Let $y^{i,i+1} = (y^{i,i})^*$ (the point at the optimum).
Let $y^{j,i+1} = y^{j,i}$ for $j \neq i$.

Step 1.2 If $i < k$ go to Step 1.1.

Step 2 Check for end conditions: if apply, stop, and return the current best solution. Otherwise, go to Step 1
BCD-CTA in $\tau$-Argus

- BCD-CTA available in $\tau$-Argus.
- Implemented within DwB EU project.
Contents

1. Introduction and motivation
2. Outline of MILP-CTA
3. BCD-CTA
4. Computational results
5. Conclusions and future work
Preliminary results

- Three-dimensional Eurostat SBS table.
- Used both MILP-CTA and BCD-CTA as implemented in $\tau$-Argus.
- CPLEX used as LP/MILP solver.
- Initially used $w_i = 1$ and MILP-CTA: stopped after 7692 seconds.
- Then we tried BCD-CTA: decent solution in 19 seconds.
- We tried other weights: unexpectedly much faster.
Summary of results

- Results for MILP-CTA and BCD-CTA with three different weights.
- $\bar{x}$ CTA: average deviation of CTA solution.
- $\bar{x}$ published: average deviation of published tables with full approach.
- BCD-CTA did a good job for CPU time and “$\bar{x}$ CTA”.
- Best combination for “$\bar{x}$ published” is likely $w_i = 1/\sqrt{a_i}$ with MILP-CTA.

<table>
<thead>
<tr>
<th>$w_i$</th>
<th>MILP-CTA</th>
<th>BCD-CTA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU</td>
<td>$\bar{x}$ CTA</td>
</tr>
<tr>
<td>1</td>
<td>7692</td>
<td>38933</td>
</tr>
<tr>
<td>$1/a_i$</td>
<td>85</td>
<td>46904</td>
</tr>
<tr>
<td>$1/\sqrt{a_i}$</td>
<td>179</td>
<td>40717</td>
</tr>
</tbody>
</table>
Graphical representation of solution

Solution with one minute of BCD-CTA followed by two hours of MILP-CTA with weights $w_i = 1$. 
Contents

1. Introduction and motivation
2. Outline of MILP-CTA
3. BCD-CTA
4. Computational results
5. Conclusions and future work
Conclusions and future work

- Cell weights have a large implication in solutions and CPU time.

- Cell weights based on cell hierarchy (J.C., S. Giessing 2006 paper).

- BCD-CTA has enormous gain in CPU time, with a price in solution.

- BCD-CTA available in \(\tau\)-Argus: it allows solution of very large and intractable tables by other approaches.

- From an optimization point of view: why the behaviour of the MILP solver changes so drastically with different weights?
Thanks for your attention!