# On using an improved Benders method for cell suppression

Work Session on Statistical Data Confidentiality
Skopje, Macedonia

D. Baena [1] J. Castro [1] A. Frangioni [2]

[1]Department of Statistics and Operations Research, Universitat Politècnica de Catalunya

[2]Department of Computer Science, Università di Pisa

September 2017

Supported by MINECO-FEDER grant MTM2015-65362-R.

# Contents

# Tabular data protection

- Statistical Disclosure Control: protect confidential information in released data.
- Two types of data:
  - Disaggregated data (or microdata): contains individual information
  - Aggregated data: tabular data by crossing categorical variables.
- Two types of methods:
  - Non-perturbative: Not change the original values, only suppress data or change the structure (Recoding or Cell Suppression Problem (CSP)).
  - Perturbative: Change the original values (Rounding, controlled rounding or Controlled Tabular Adjustment (CTA)).

# CSP: Cell Suppression problem

- Find the minimum set of additional cells to be suppressed so that the value of sensitive cells cannot be recompiled.
- Example of cell suppression (Protection levels: $upl_{23} = lpl_{23} = 10$):

| | Original Table | | | |
|---|---|---|---|---|
| | $t_1$ | $t_2$ | $t_3$ | **Total** |
| $e_1$ | 20 | 24 | 28 | **72** |
| $e_2$ | 38 | 38 | **40** | **116** |
| $e_3$ | 40 | 39 | 42 | **121** |
| **Total** | **98** | **101** | **110** | **309** |

| | Published table | | | |
|---|---|---|---|---|
| | $t_1$ | $t_2$ | $t_3$ | **Total** |
| $e_1$ | 20 | 24 | 28 | **72** |
| $e_2$ | 40 | 38 | | **116** |
| $e_3$ | 40 | 39 | 42 | **121** |
| **Total** | **98** | **101** | **110** | **309** |

- A lower $\underline{a_s}$ and upper $\overline{a_s}$ bound can be obtained for each sensitive cell $s$.
- Published table is safe if and only if for each sensitive cell $s$:
  - $\underline{a_s} \leq a_s - lpl_s$ and $\overline{a_s} \geq a_s + upl_s$
  - In the example: $\underline{a_{23}} = 20 \leq 40 - 10$ and $\overline{a_{23}} = 68 > 40 + 10$

# CSP: Cell Suppression problem

- Find the minimum set of additional cells to be suppressed so that the value of sensitive cells cannot be recompiled.
- Example of cell suppression (Protection levels: $upl_{23} = lpl_{23} = 10$):

| Original Table | $t_1$ | $t_2$ | $t_3$ | **Total** |
|---|---|---|---|---|
| $e_1$ | 20 | 24 | 28 | **72** |
| $e_2$ | 38 | 38 | 40 | **116** |
| $e_3$ | 40 | 39 | 42 | **121** |
| **Total** | **98** | **101** | **110** | **309** |

| Published table | $t_1$ | $t_2$ | $t_3$ | **Total** |
|---|---|---|---|---|
| $e_1$ | | 24 | | **72** |
| $e_2$ | | 38 | | **116** |
| $e_3$ | 40 | 39 | 42 | **121** |
| **Total** | **98** | **101** | **110** | **309** |

- A lower $\underline{a_s}$ and upper $\overline{a_s}$ bound can be obtained for each sensitive cell $s$.
- Published table is safe if and only if for each sensitive cell $s$:
  - $\underline{a_s} \leq a_s - lpl_s$ and $\overline{a_s} \geq a_s + upl_s$
  - In the example: $\underline{a_{23}} = 20 \leq 40 - 10$ and $\overline{a_{23}} = 68 > 40 + 10$

# The MILP CSP model

- The formulation of CSP is a very large MILP problem.
- For instance: a table with 4.000 cells ($n$), 1.000 sensitive cells ($\mathcal{S}$) and 2.500 linear relations ($m$) formulates a MILP with:
  - 4.000 binary variables ($n$).
  - 8.000.000 continuous variables ($2n|\mathcal{S}|$).
  - 21.000.000 constraints ($2(m+2n)|\mathcal{S}|$).
- CSP becomes impractical for state-of-the-art MILP solvers.
- CSP is solved using Benders decomposition method. Implemented in Tau-Argus (SDC software).

# Benders decomposition

- Benders decomposition is an iterative method which decomposes the original MILP in several smaller subproblems easier to solve.
  - Relaxed Master problem in only the "complicating" binary variables.
  - Subproblems on the "easy" continuous variables (with the binary variables fixed).
  - At each iteration: optimality and feasibility cuts generated by subproblems added to the relaxed Master.
- It is widely used in many real-world applications. However...
  - suffers from well known instability issues.
  - solutions jump from a "good" point to a one worse.
  - The convergence to the optimum is often too slow.
- Goal: To improve the Benders decomposition with the application of stabilization techniques.

# Stabilized Benders methods

- **Idea**: Focus on finding new solutions in the neighborhood of "good" points.
- For binary MILPs, the stabilization can be done by adding linear constraints that:
  - Restrict the feasible region of relaxed master problems.
    - ★ Trust region (TR).
  - How?: Using the Hamming distance defined from:
    - ★ A stability center point ("good" point).
    - ★ A radius $K \geq 1$.
- In the case of binary variables, the trust region (TR) is defined using local branching constraints (LB)

# Benefits

- The main benefits of stabilization techniques are:
  - Reduction of the total computational time because fewer iterations are required.
  - Relaxed master problems of smaller feasible region and theoretically easier to solve.
  - The search for solutions around a good point considered increases the chance of finding better feasible solutions.

# Computational results

- 48 randomly 1H2D asymmetric tables:

| Asymmetric instances | |
|---|---|
| **Stabilized Benders CSP** | **State-of-the-art classical Benders CSP** |
| Average gap 0.87% | Average gap 2.51% |
| 1.8 times faster | |
| | In 19% didn't find feasible solution * |
| Better in 92% of instances | |

* within the time limit (3600 seconds).

- 15 real-world tables:

| Real-world general instances | |
|---|---|
| **Stabilized Benders CSP** | **State-of-the-art classical Benders CSP** |
| Better in 53% of instances (average gap 2.61%) | |
| | Better in 47% of instances (average gap 4.57%) |
| | Not feasible solution in 2 instances |

# Computational results (I)

- Benders algorithm built in CPLEX 12.7 (CPLEX-Benders) vs stabilized Benders in a set of small random 1H2D tables.
- For this test CPLEX was interfaced through AMPL (we only considered the CPLEX solution time).

| | Stabilized Benders | | CPLEX-Benders | |
|---|---|---|---|---|
| | **CPU time** | **gap** | **CPU time** | **gap** |
| 20_25_15_5_1 | **51,44** | **0,01%** | 3596.43 | 0,09% |
| 20_30_15_5_1 | **2297,58** | **0,01%** | 3596.50 | 1,05% |
| 20_35_15_5_1 | **163,14** | **0,01%** | 3596,39 | 0,77% |
| 25_25_15_5_1 | **3600,11** | **0,03%** | 3596,32 | 0,35% |
| 25_30_15_5_1 | **54,87** | **0,01%** | 3596,5 | 5,68% |
| 25_35_15_5_1 | **43,61** | **0,00%** | 3596,55 | 94,98% |
| 30_25_15_5_1 | **1128,73** | **0,01%** | 3596,38 | 0,33% |
| 30_30_15_5_1 | **99,76** | **0,01%** | 3596,56 | 94,91% |
| 30_35_15_5_1 | **568,42** | **0,01%** | ‡ | ‡ |

‡ internal memory error provided by AMPL;

- **Benders algorithm built in CPLEX 12.7 is not competitive with stabilized Benders.**

# Conclusions

- Stabilized Benders applied to CSP: Excellent strategy compared to the state-of-the-art classical Benders.
- In 92% of the synthetic 1H2D tables: stabilized Benders outperformed classical Benders (both CPU and gap).
- For real-world general tables the stabilized strategy was not as competitive as for 1H2D case:
  - Probably due to the absence of a hierarchical structure.
  - However, it can be a promising approach: the average gap was lower (2.61%) than for state-of-the-art classical Benders
- D.Baena, J.Castro, A. Frangioni, Stabilized Benders methods for large combinatorial optimization problems: applications to cell suppression, working paper to be submitted.

**Thanks**