"A Matheuristic for a Customer Assignment Problem in Direct Marketing"

T. Bigler, M. Kammermann, P. Baumann

European Journal of Operational Research, 2022

Spreafichi Frederick

April 27, 2025

Index

Business Context and Problem

- A telecommunications company runs **multiple direct marketing campaigns** targeting different customer segments
- Marketing managers design offers, select eligible customers, and define **campaign-specific constraints**
- Central management defines **global constraints** like budgets
- **Customer assignment** is centralized to avoid over-contact and ensure all constraints are respected
- The company requested the development of a fast and reliable heuristic to automate the assignment process

Problem Description

- **Input:**
  - **I**: set of customers
  - **J**: set of activities
  - $\mathbf{e_{ij}}$: expected profit when customer i is assigned to activity j
  - $\mathbf{q_{ij}}$: response probability
  - $\mathbf{c_j}$: cost of activity j
  - Campaigns' time schedules
  - Business constraints
  - Customer-specific constraints
- **Objective:**
  - Assign customers to activities to **maximize** the total expected profit, subject to the given constraints

Problem Description

Business constraints:

- Minimum assignment (soft constraint)
- Maximum assignment
- Budget
- Minimum sales (soft constraint)
- Maximum sales (soft constraint)

Customer-specific constraints:

- Minimum contact (soft constraint)
- Maximum contact
- Conflict constraints

Comparison with other combinatorial optimization problems

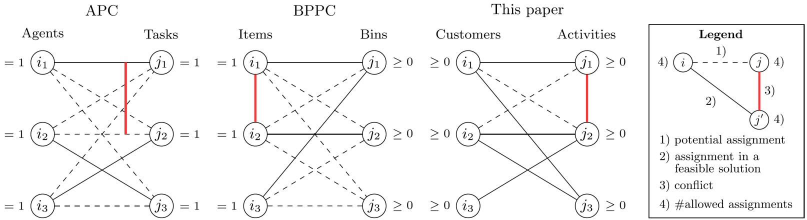


Fig. 2. Conflict constraints in more general combinatorial optimization problems.

Mixed Binary Linear Program (Notations)

- $\mathbf{x_{ij}} \in \{0,1\}$: Decision variable indicating whether customer *i* is assigned to activity *j*
- z_l^a, z_l^s, $z_l^{\bar{s}}$, z_l^m: Slack variables for the minimum assignment, minimum sales, maximum sales, and minimum contact constraints, respectively
- $\mathbf{J_l^a}$, $\mathbf{J_l^{\bar{a}}}$, $\mathbf{J_l^b}$, $\mathbf{J_l^s}$, $\mathbf{J_l^{\bar{s}}}$, $\mathbf{J_l^m}$, $\mathbf{J_l^{\bar{m}}}$: Subsets of activities associated with each constraint type
- $\mathbf{b_l^a}$, $\mathbf{b_l^s}$, $\mathbf{b_l^{\bar{s}}}$, $\mathbf{b_l^m}$, $\mathbf{b_l^{\bar{m}}}$: Bounds for the respective constraints
- α, β, γ, δ: Penalty constants for the respective soft constraints

Mixed Binary Linear Program

$$(1) \quad \max \sum_{j\in J}\sum_{i\in I_j} e_{ij}x_{ij} - \left(\alpha \sum_{l=1}^{n_a} z_l^a + \beta \sum_{l=1}^{n_s} z_l^s + \gamma \sum_{l=1}^{n_{\bar{s}}} z_l^{\bar{s}} + \delta \sum_{i\in I}\sum_{l=1}^{n_m} z_{il}^m \right) \quad \textbf{s.t.}$$

(2)	$\sum_{j\in J_l^a}\sum_{i\in I_j} x_{ij} + z_l^a \geq b_l^a$	(Min assignment)	$z_l^a \in [0, b_l^a]$
(3)	$\sum_{j\in J_l^{\bar{a}}}\sum_{i\in I_j} x_{ij} \leq b_l^{\bar{a}}$	(Max assignment)	
(4)	$\sum_{j\in J_l^b}\sum_{i\in I_j} c_j x_{ij} \leq b_l^b$	(Budget)	
(5)	$\sum_{j\in J_l^s}\sum_{i\in I_j} q_{ij}x_{ij} + z_l^s \geq b_l^s$	(Min sales)	$z_l^s \in [0, b_l^s]$
(6)	$\sum_{j\in J_l^{\bar{s}}}\sum_{i\in I_j} q_{ij}x_{ij} - z_l^{\bar{s}} \leq b_l^{\bar{s}}$	(Max sales)	$z_l^{\bar{s}} \in [0, \bar{q}]$
(7)	$\sum_{j\in J_l^m \cap J_i} x_{ij} + z_{il}^m \geq b_l^m$	(Min contact)	$z_{il}^m \in [0, b_l^m]$
(8)	$\sum_{j\in J_l^{\bar{m}} \cap J_i} x_{ij} \leq b_l^{\bar{m}}$	(Max contact)	
(9)	$x_{ij_1} + x_{ij_2} \leq 1$	(conflict)	

Where $\quad \bar{q} = \sum_{i\in I}\sum_{j\in J} q_{ij}$

Matheuristic Overview

- Instead of modeling individual customer assignments, **the matheuristic solves a model for groups of customers**
- It **incorporates customer-specific constraints** already at the group level
- A single parameter controls the **trade-off** between **solution quality** and **running time**
- The approach follows **four steps**:
  1. Group customers by common eligibility
  2. Cluster groups based on expected profit
  3. Solve a group-level linear program
  4. Iteratively assign individual customers

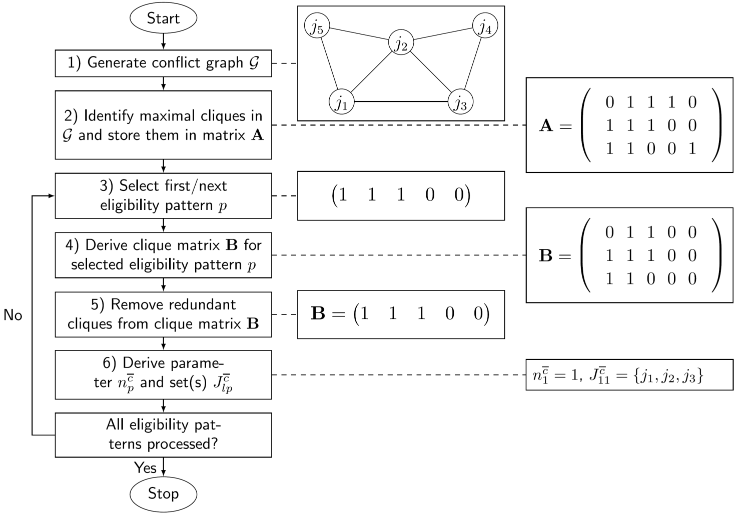
Matheuristic: Step 1 - Grouping by Eligibility Patterns

- Customers are grouped based on their **eligibility pattern p**, indicating the set of activities they are eligible for
- Eligibility pattern is a binary vector (e.g., `[1, 0, 1]`), where '1' indicates eligibility for the activity
- Grouping is essential as it:
  - **Simplifies the decomposition strategy**
  - Ensures customers in the same group are affected by the **same conflicts**
  - Allows **enforcing customer-specific constraints** later

Matheuristic: Step 2 - Clustering Subgroups

- Each group is divided into up to **k** subgroups
- Customers in the same subgroup have **similar expected profits** for eligible activities
- **Mini-batch *k*-means** algorithm is used for efficiency and scalability
- Parameter k allows controlling the **trade-off** between **solution quality** and **computational time**

Preprocessing Technique



Preprocessing: Redundant Cliques Removal (step 5)

$$\mathbf{B} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix} \xrightarrow{①} \mathbf{B}(\mathbf{B})^T = \mathbf{C} = \begin{pmatrix} 2 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 2 \end{pmatrix}, \mathbf{D} = \begin{pmatrix} 2 & 2 & 2 \\ 3 & 3 & 3 \\ 2 & 2 & 2 \end{pmatrix} \xrightarrow{②} \mathbf{C} - \mathbf{D} = \mathbf{E} =$$

$$\begin{pmatrix} 0 & 0 & -1 \\ -1 & 0 & -1 \\ -1 & 0 & 0 \end{pmatrix} \xrightarrow{③} \mathbf{E} = \begin{pmatrix} -1 & 0 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \end{pmatrix} \xrightarrow{④} \mathbf{B} = \begin{pmatrix} \cancel{0} & \cancel{1} & \cancel{1} & \cancel{0} & \cancel{0} \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} \cancel{2} & \cancel{2} & \cancel{1} \\ \cancel{2} & 3 & 2 \\ \cancel{1} & 2 & 2 \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ 2 & 2 \end{pmatrix}, \mathbf{D} = \begin{pmatrix} \cancel{2} & \cancel{2} & \cancel{2} \\ \cancel{3} & 3 & 3 \\ \cancel{2} & 2 & 2 \end{pmatrix} = \begin{pmatrix} 3 & 3 \\ 2 & 2 \end{pmatrix} \xrightarrow{⑤} \mathbf{C} - \mathbf{D} = \mathbf{E} = \begin{pmatrix} 0 & -1 \\ 0 & 0 \end{pmatrix}$$

$$⑥\ \mathbf{E} = \begin{pmatrix} -1 & -1 \\ 0 & -1 \end{pmatrix} \xrightarrow{⑦} \mathbf{B} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ \cancel{1} & \cancel{1} & \cancel{0} & \cancel{0} & \cancel{0} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Fig. 7. Example for removing cliques from the binary matrix **B** that are a subset of another clique.

Matheuristic: Step 3 - Linear Program (Notations)

- $\mathbf{G}$: Subgroups determined in step 2
- $\mathbf{x_{gj}} \in [0, o_g]$: Decision variables. Number of customer of group g assigned to activity j
- $\mathbf{o_g}$: Number of customers in group g
- $\mathbf{\bar{e}_{gj}}$: Average expected profit of customers of group g when assigned to activity j
- $\mathbf{\bar{q}_{gj}}$: Average response probability of customers of group g when assigned to activity j
- $\mathbf{z_{gl}^{m}}$: Slack variable of group g for minimum contact rule

Matheuristic: Step 3 - Linear Program

$$(15) \quad \max \sum_{g \in G} \sum_{j \in J_g} \bar{e}_{gj} x_{gj} - \left(\alpha \sum_{l=1}^{n_a} z_l^a + \beta \sum_{l=1}^{n_s} z_l^s + \gamma \sum_{l=1}^{n_{\bar{s}}} z_l^{\bar{s}} + \delta \sum_{g \in G} \sum_{l=1}^{n_m} z_{gl}^m \right) \quad \textbf{s.t.}$$

(16)	$\sum_{g \in G} \sum_{j \in J_l^a \cap J_g} x_{gj} + z_l^a \geq b_l^a$	(Min assignment)	$z_l^a \in [0, b_l^a]$
(17)	$\sum_{g \in G} \sum_{j \in J_l^{\bar{a}} \cap J_g} x_{gj} \leq b_l^{\bar{a}}$	(Max assignment)	
(18)	$\sum_{g \in G} \sum_{j \in J_l^b \cap J_g} c_j x_{gj} \leq b_l^b$	(Budget)	
(19)	$\sum_{g \in G} \sum_{j \in J_l^s \cap J_g} \bar{q}_{gj} x_{ij} + z_l^s \geq b_l^s$	(Min sales)	$z_l^s \in [0, b_l^s]$
(20)	$\sum_{g \in G} \sum_{j \in J_l^{\bar{s}} \cap J_g} \bar{q}_{gj} x_{ij} - z_l^{\bar{s}} \leq b_l^{\bar{s}}$	(Max sales)	$z_l^{\bar{s}} \in [0, \bar{q}]$
(21)	$\sum_{j \in J_l^m \cap J_g} x_{gj} + z_{gl}^m \geq o_g b_l^m$	(Min contact)	$z_{gl}^m \in [0, o_g b_l^m]$
(22)	$\sum_{j \in J_l^{\bar{m}} \cap J_g} x_{gj} \leq o_g b_l^{\bar{m}}$	(Max contact)	
(23)	$\sum_{j \in J_{lp}^c} x_{gj} \leq o_g$	(Conflict)	

Iterative Algorithm

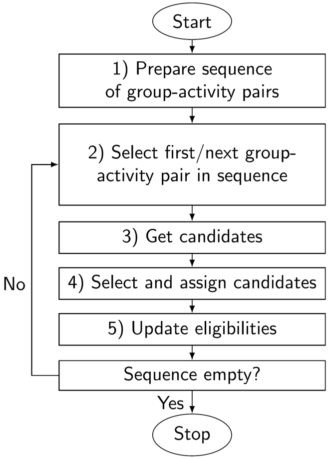


Fig. 4. Flowchart of the iterative algorithm.

Alternative MBLP

- **Goal:** Reduce number of conflict constraints by using maximal sets of conflicting activities from the preprocessing step
- **Conflict Constraint Formulation:**

$$(26) \quad \sum_{j \in J^c_{lp}} x_{ij} \leq 1 \quad \text{for all } i \in I_p,\ l = 1, \ldots, n^p_c$$

- **Benefits:**
  - Fewer constraints than original MBLP
  - Tighter LP relaxation
  - More scalable for larger instances

Datasets Overview

Generated Instances:

- **GS (Small)**: up to 20,000 customers and 75 activities
- **GM (Medium)**: up to 200,000 customers and 125 activities
- *GL (Large)*: up to 1,000,000 customers and 175 activities

Real-World Instances:

- *RL (Large)*: up to 1.4M customers and 385 activities
- *RVL (Very Large)*: over 2M customers and 295 activities

Datasets vary in terms of **eligibility fraction** and **number of eligibility patterns**

Experimental Setup

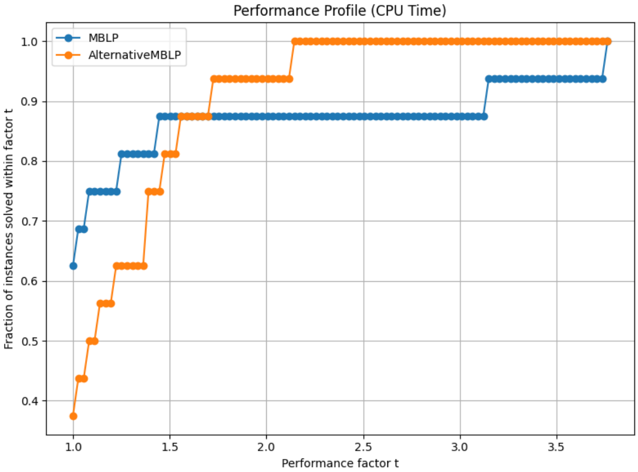
- **Solver:** Gurobi 11.0
- **Programming Language:** Python 3.12
- **Time Limit:** 30 minutes (company requirement)
- **Matheuristic parameter k:** Default $k = 20$, tested with varying k
- **Penalty constants $\alpha, \beta, \gamma, \delta$:** Set to maximum expected profit
- **Note:** Import/export time excluded as identical across models

Scalability Analysis: MBLP vs MBLP'

	MBLP						ALTERNATIVE MBLP					
	OFV (100k)	PENALTY (100K)	MIPGAP (%)	CONSTR. (1K)	CPU (SEC)	STATUS	OFV (100k)	PENALTY (100K)	MIPGAP (%)	CONSTR. (1K)	CPU (SEC)	STATUS
GM1	34,1	0.0	0.0	1144	87.62	Optimal	34,1	0.0	0.0	277	98.12	Optimal
GM2	84,7	0.0	0.0	5918	369.74	Optimal	84,7	0.0	0.0	539	259.02	Optimal
GM3	24,6	0.0	0.0	1232	86.56	Optimal	24,6	0.0	0.0	287	118.87	Optimal
GM4	29,7	0.0	0.0	5742	250.83	Optimal	29,7	0.0	0.0	534	204.42	Optimal
GM5	45,3	0.0	0.0	3138	232.57	Optimal	45,3	0.0	0.0	624	235.72	Optimal
GM6	-149,4	-	-	18118	-	Out of Memory	144,8	0.0	0.0	1243	2523.54	Optimal
GM7	68,4	0.0	0.0	3264	249.46	Optimal	68,4	0.0	0.0	621	267.28	Optimal
GM8	-	-	-	-	-	Out of Memory	128,3	0.0	0.0	1225	2800.09	Optimal

- Both models solve to optimality the instances they are capable of handling
- **Alternative MBLP** features **fewer constraints** and demonstrates **better scalability**

Scalability Analysis: Performance Profile



Scalability Analysis: MBLP' vs Matheuristic

	ALTERNATIVE MBLP						MATHEURISTIC					
	OFV (100k)	PENALTY (100K)	MIPGAP (%)	CONSTR. (1K)	CPU (SEC)	STATUS	OFV (100k)	PENALTY (100K)	MIPGAP (%)	CONSTR. (1K)	CPU (SEC)	K
GM1	34,1	0.0	0.0	277	98.12	Optimal	33,5	0.0	**1.69**	**17**	**24.67**	20
GM2	84,7	0.0	0.0	539	259.02	Optimal	80,7	0.0	**4.69**	**32**	**33.13**	20
GM3	24,6	0.0	0.0	287	118.87	Optimal	24,4	0.0 (1)	**0.92**	**48**	**54.52**	20
GM4	29,7	0.0	0.0	534	204.42	Optimal	28,9	0.0 (1)	**2.82**	**85**	**70.33**	20
GM5	45,3	0.0	0.0	624	235.72	Optimal	44,5	0.0 (1)	**1.77**	**19**	**38.55**	20
GM6	144,8	0.0	0.0	1243	2523.54	Optimal	138,8	0.0	**4.14**	**37**	**57.46**	20
GM7	68,4	0.0	0.0	621	267.28	Optimal	67,3	0.0 (1)	**1.62**	**51**	**66.47**	20
GM8	128,3	0.0	0.0	1225	2800.09	Optimal	121,8	0.0 (1)	**5.12**	**98**	**100.26**	20

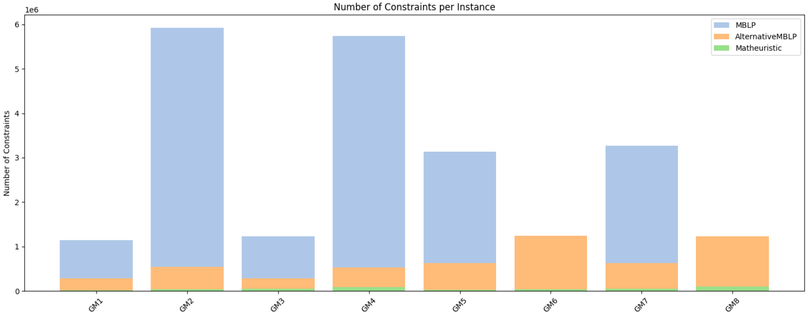
- The **matheuristic** drastically **reduces the number of constraints** and consistently achieves much **faster solving times** across all instances
- The **MipGap remains contained** and never exceeds 5.12% even with fixed k=20

Scalability Analysis: Models Constraints

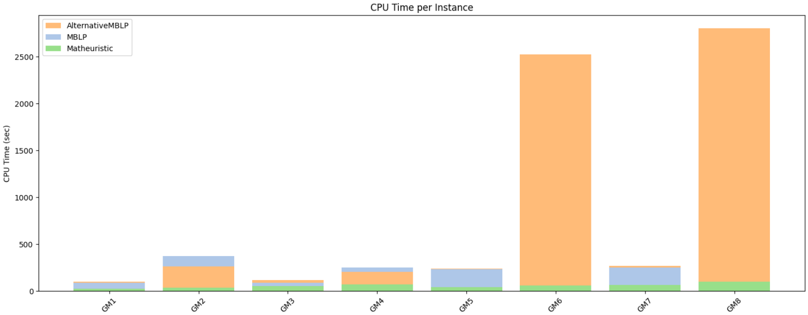
Dataset	Customers (1k)	Activities	Eligibility Fraction	Eligibility Patterns	MBLP	AlternativeMBLP	Matheuristic (conflict constrs.)
GS1	10	50	small (5)	few (50)	34	13	2 (1)
GS2	**10**	**50**	**large (15)**	few (50)	**176**	**35**	**3 (2)**
GS3	10	50	small (5)	**many (100)**	41	16	4 (2)
GS4	**10**	**50**	**large (15)**	**many (100)**	**182**	**35**	**7 (5)**

- The **eligibility fraction** influences the number of constraints much more significantly than the number of eligibility patterns

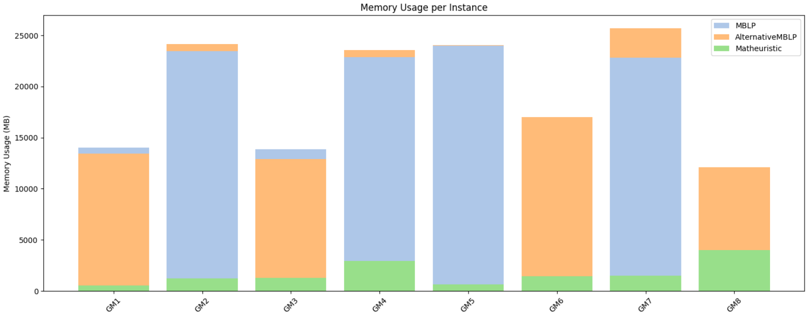
Scalability Analysis: Number of Constraints - GM Datasets



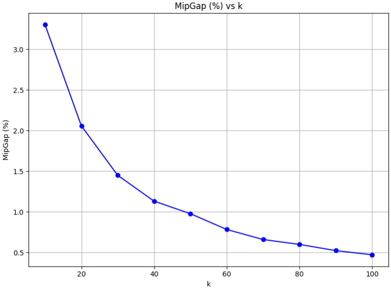
Scalability Analysis: CPU Time (sec) - GM Datasets



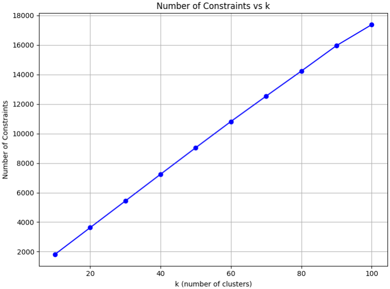
Scalability Analysis: Memory (MB) - GM Datasets



Scalability Analysis: k Values



MIP Gap vs. k



Number of Constraints vs. k

We now analyze:

- **Impact of Step 5 of the preprocessing technique:**
  Effect of removing redundant subcliques at the group-level on matheuristic performance

- **Impact of the new modeling technique:**
  Effect of the group-level approach to conflict constraints on matheuristic performance

Matheuristic without the New Modeling Technique

Benchmark version of the matheuristic:

- Conflict constraints are included in the linear program by directly formulating **constraint (9) of the MBLP at the group level**
- The resulting linear program (LP) is defined as:
  - **Objective:** Maximize total expected profit (equation 15)
  - **Subject to:** Constraints (11)–(13), (16)–(22), (24), (25)
  - **Conflict constraints:**

$$x_{gj_1} + x_{gj_2} \leq o_g \quad (g \in G;\ (j_1, j_2) \in T:\ j_1, j_2 \in J_g)$$

Matheuristic: New Modeling Technique Impact

	MATHEURISTIC					MATHEURISTIC WITHOUT NEW MODELING TECHNIQUE				
	OFV (100k)	PENALTY (100K)	MIPGAP (%)	CONSTR. (1K)	CPU (SEC)	OFV (100k)	PENALTY (100K)	MIPGAP (%)	CONSTR. (1K)	CPU (SEC)
GM1	33,5	0.0	**1.69**	**17**	24.67	26,2	3.3	**22.96**	**69**	31.55
GM2	80,7	0.0	**4.69**	**32**	33.13	46,9	17.3	**44.61**	**355**	47.96
GM3	24,4	0.0 (1)	**0.92**	**48**	54.52	-15,1	35.3	**161.16**	**199**	71.81
GM4	28,9	0.0 (1)	**2.82**	**85**	70.33	-84,8	107.3	**385.08**	**915**	109.52
GM5	44,5	0.0 (1)	**1.77**	**19**	38.55	36,8	1.3	**18.72**	**94**	51.94
GM6	138,8	0.0	**4.14**	**37**	57.46	58,4	52.2	**139.12**	**543**	82.84
GM7	67,3	0.0 (1)	**1.62**	**51**	66.47	50,4	6.4	**26.3**	**262**	102.68
GM8	121,8	0.0 (1)	**5.12**	**98**	100.26	58,1	36.1	**54.7**	**1446**	169.16

	Matheuristic		Matheuristic without new modeling	
	LP #args	Iterative alg # args (difference)	LP #args	Iterative alg #args (difference)
∑ GS	156105	156090 **(-0.01%)**	182447	144743 **(-20.67%)**
∑ GM	2203748	2203719 **(0%)**	2863580	1862789 **(-34.94%)**

Matheuristic: Preprocessing Step 5 impact

		Matheuristic	Matheuristic without step 5
Number of Constraints	∑ GS	34	294 **(+764%)**
	∑ GM	387	6507 **(+1581%)**
Number of Conflict Constraints	∑ GS	22	282 **(+1181%)**
	∑ GM	299	6419 **(+2046%)**

- **Step 5** is critical for **reducing** the number of constraints and in particular the number of **conflict constraints**

Matheuristic: Performance Overview

- **Significantly faster** than the other analyzed models
- **Near-optimal solutions** achieved in most cases with $k = 20$
- Increasing k improves solution quality, with a faster-than-linear decrease in the optimality gap
- **Average decrease of 97%** in the number of constraints
- Solution quality is **more sensitive to the eligibility fraction** than to the number of patterns
- **Excellent scalability**: maintains high solution quality as instance size grows

Conclusions

- **Context:** A real-world planning problem from a telecommunications company, involving customer assignments to direct marketing activities under complex business and customer-specific constraints
- **Approach:** Development of a matheuristic combining a group-level linear model with an iterative customer-level assignment
- **Key Techniques:** Introduction of new modeling and preprocessing methods to efficiently manage customer-specific conflict constraints
- **Real-world Impact:** Adoption by the company, leading to a **90% increase in sales** and a **300% improvement in campaign profitability**

References

Paper reference:

T. Bigler, M. Kammermann, P. Baumann

"A matheuristic for a customer assignment problem in direct marketing"

Published in **European Journal of Operational Research**

Volume 304, 2023, Pages 689–708

DOI: 10.1016/j.ejor.2022.04.009

GitHub repository with generated instances:

https://github.com/phil85/customer-assignment-instances