"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 14, 2025

Index

Problem Description

Business context:

A telecommunications company runs multiple direct marketing campaigns targeting different customer segments. Campaign managers define offers, target products, and campaign-specific constraints, while central management sets global constraints like budgets.

A central unit manually assigns customers to activities, ensuring constraint compliance and avoiding over-contact. However, with millions of customers and hundreds of activities, the process leads to hundreds of millions of conflict constraints.

Due to the problem size and the lack of suitable heuristics, an ad hoc heuristic approach is required.

Problem Description

- **Input:**
  - **I**: set of customers
  - **J**: set of activities
  - e_{ij}: expected profit when customer i is assigned to activity j
  - q_{ij}: response probability
  - c_j: cost of activity j
  - Campaign start and end dates
  - Scheduled day of each activity
  - Business constraints
  - Customer specific constraints

- **Objective:**
  - Assign customers to activities to **maximize** 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)
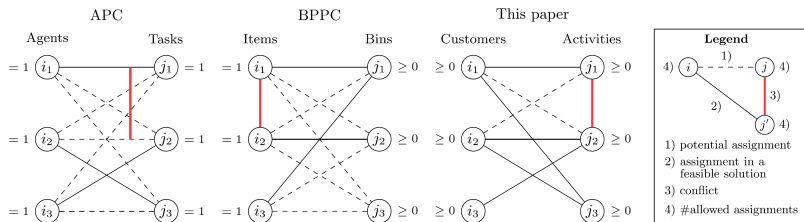- Maximum contact
- Conflict constraints

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

Mixed Binary Linear Program (Notations)

- **I**: Set of customers

- **J**: Set of activities

- e_{ij}: Expected profit when customer i is assigned to activity j

- q_{ij}: Response probability of customer i for activity j

- c_j: Cost associated with activity j

- $x_{ij} \in \{0,1\}$: Decision variable indicating whether customer i is assigned to activity j

- I_j: Set of customers eligible for activity j

- J_i: Set of activities for which customer i is eligible

- $\alpha,\ \beta,\ \gamma,\ \delta$: Penalty constants for the respective soft constraints

- $z_i^a,\ z_i^s,\ z_i^{\bar{s}},\ z_i^m$: Slack variables for the minimum assignment, minimum sales, maximum sales, and minimum contact constraints, respectively

- $J_i^a,\ J_i^{\bar{a}},\ J_i^b,\ J_i^s,\ J_i^{\bar{s}},\ J_i^m,\ J_i^{\bar{m}}$: Subsets of activities associated with each constraint type

- $b_i^a,\ b_i^s,\ b_i^{\bar{s}},\ b_i^m,\ b_i^{\bar{m}}$: Bounds for the respective constraints

Mixed Binary Linear Program

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

(2) $\displaystyle \sum_{j \in J_l^a} \sum_{i \in I_j} x_{ij} + z_l^a \geq b_l^a$ (Min assignment) $z_l^a \in [0, b_l^a]$

(3) $\displaystyle \sum_{j \in J_l^{\bar{a}}} \sum_{i \in I_j} x_{ij} \leq b_l^{\bar{a}}$ (Max assignment)

(4) $\displaystyle \sum_{j \in J_l^b} \sum_{i \in I_j} c_j x_{ij} \leq b_l^b$ (Budget)

(5) $\displaystyle \sum_{j \in J_l^s} \sum_{i \in I_j} q_{ij} x_{ij} + z_l^s \geq b_l^s$ (Min sales) $z_l^s \in [0, b_l^s]$

(6) $\displaystyle \sum_{j \in J_l^{\bar{s}}} \sum_{i \in I_j} q_{ij} x_{ij} - z_l^{\bar{s}} \leq b_l^{\bar{s}}$ (Max sales) $z_l^{\bar{s}} \in [0, \bar{q}]$

(7) $\displaystyle \sum_{j \in J_l^m \cap J_i} x_{ij} + z_{il}^m \geq b_l^m$ (Min contact) $z_{il}^m \in [0, b_l^m]$

(8) $\displaystyle \sum_{j \in J_l^{\bar{m}} \cap J_i} x_{ij} \leq b_l^{\bar{m}}$ (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** at the group level, enabling the group-level solution to be efficiently transformed into a near-optimal customer-level assignment

- A single parameter lets the user adjust 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

- 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 means eligible for the activity

- Customers sharing the same pattern are placed in the same group

- **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

- In practice, the number of eligibility patterns is much smaller than the number of customers, keeping the approach efficient

- Each group is divided into up to **k** subgroups

- Customers in the same subgroup have **similar expected profits** for eligible activities

- **Input matrix**: rows are customers, columns are eligible activities, values are expected profits

- **Mini-batch k-means** algorithm is used for efficiency and scalability

- Increasing k leads to smaller and more homogeneous subgroups

- Parameter k allows controlling the **trade-off** between **solution quality** and **computational time**

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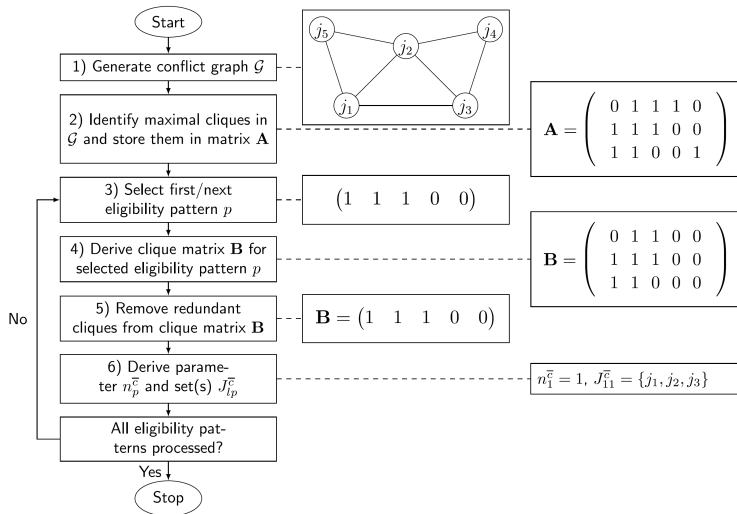
$$\mathbf{B} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix} \xrightarrow{\text{①}} \mathbf{B}(\mathbf{B})^T = \mathbf{C} = \begin{pmatrix} 2 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 2 \end{pmatrix}, \mathbf{D} = \begin{pmatrix} 2 & 2 & 2 \\ 3 & 3 & 3 \\ 2 & 2 & 2 \end{pmatrix}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

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

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

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

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

Matheuristic: Step 3 - Linear Program (Notations)

- **G**: Subgroups determined in step 2

- **P**: Eligibility patterns

- $\mathbf{x_{gj}} \in [0, o_g]$: Decision variables. Is the number of customer of group g that are assigned to activity j

- $\mathbf{o_g}$: Number of customers in group g

- $\mathbf{J_g}$: Activities for which customers in group g are eligible

- $\mathbf{J_{lp}^c}$: Activities associated with constraint $l = 1, ..., n_p^c$ which ensures conflict rules for groups with eligibility pattern p

- $\mathbf{\bar{e}_{gj}}$: 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) $\max \sum_{g \in G} \sum_{j \in J_g} \bar{e}_{gj} x_{gj} - \left(\alpha \sum_{l=1}^{n_a} z_l^a + \beta \sum_{l=1}^{n_s} z_l^s + \gamma \sum_{l=1}^{n_{\bar{s}}} z_l^{\bar{s}} + \delta \sum_{g \in G} \sum_{l=1}^{n_m} z_{gl}^m \right)$ **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
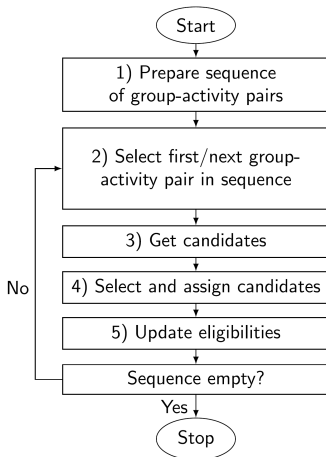
Fig. 4. Flowchart of the iterative algorithm.

Alternative MBLP

- **Goal:** Reduce number of conflict constraints by using conflict sets for subgroups

- **Conflict Constraint Formulation:**

$$(26) \quad \sum_{j \in J_{lp}^c} x_{ij} \leq 1 \quad \text{for all } i \in I_p, \ l = 1, \ldots, n_c^p$$

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

- **Conclusion:** Effective alternative to original MBLP, preserving feasibility while improving computational efficiency

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

Note: Focus is on **GS** and **GM** instances due to computational limits
Datasets vary by **eligibility fraction** and **number of eligibility patterns**

Experimental Setup

- **Operating System:** Windows 11
- **Processor:** Intel Core i7-12700KF (12th Gen) @ 3.61 GHz
- **Memory:** 32 GB RAM
- **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'

Dataset	MBLP							ALTERNATIVE MBLP						
	OFV (100k)	PENALTY (100K)	MIPGAP (%)	CONSTR. (1K)	MEMORY (MB)	CPU (SEC)	STATUS	OFV (100k)	PENALTY (100K)	MIPGAP (%)	CONSTR. (1K)	MEMORY (MB)	CPU (SEC)	STATUS
GS1	1,3	0.0 (1)	0.0	34	665	3.56	Optimal	1,3	0.0 (1)	0.0	13	654	6.14	Optimal
GS2	1,2	0.0	0.0	176	1330	8.56	Optimal	1,2	0.0	0.0	35	1227	10.29	Optimal
GS3	0,98	0.0	0.0	41	695	3.89	Optimal	0,98	0.0	0.0	16	682	8.27	Optimal
GS4	4,7	0.0	0.0	182	1507	9.11	Optimal	4,7	0.0	0.0	35	1370	12.61	Optimal
GS5	1,2	0.0	0.0	36	1548	6.79	Optimal	1,2	0.0	0.0	20	1551	9.95	Optimal
GS6	13,3	0.0	0.0	805	4572	32.32	Optimal	13,3	0.0	0.0	83	3987	29.95	Optimal
GS7	3,5	0.2 (1)	0.0	72	1787	9.22	Optimal	3,5	0.2 (1)	0.0	29	1784	14.3	Optimal
GS8	-16,3	23.0 (2)	0.0	729	4484	31.67	Optimal	-16,3	23 (2)	0.0	81	3965	31.39	Optimal
GM1	34,1	0.0	0.0	1144	14011	87.62	Optimal	34,1	0.0	0.0	277	13413	98.12	Optimal
GM2	84,7	0.0	0.0	5918	23481	369.74	Optimal	84,7	0.0	0.0	539	24162	259.02	Optimal
GM3	24,6	0.0	0.0	1232	13884	86.56	Optimal	24,6	0.0	0.0	287	12883	118.87	Optimal
GM4	29,7	0.0	0.0	5742	22889	250.83	Optimal	29,7	0.0	0.0	534	23582	204.42	Optimal
GM5	45,3	0.0	0.0	3138	23997	232.57	Optimal	45,3	0.0	0.0	624	24040	235.72	Optimal
GM6	-149,4	-	-	18118	-	-	Out of Memory	144,8	0.0	0.0	1243	16982	2523.54	Optimal
GM7	68,4	0.0	0.0	3264	22819	249.46	Optimal	68,4	0.0	0.0	621	25674	267.28	Optimal
GM8	-	-	-	-	-	-	Out of Memory	128,3	0.0	0.0	1225	12103	2800.09	Optimal

Scalability Analysis: Performance Profile

Performance Profile (CPU Time)

|	ALTERNATIVE MBLP							MATHEURISTIC						
	OFV (100k)	PENALTY (100K)	MIPGAP (%)	CONSTR. (1K)	MEMORY (MB)	CPU (SEC)	STATUS	OFV (100k)	PENALTY (100K)	MIPGAP (%)	CONSTR. (1K)	MEMORY (MB)	CPU (SEC)	K
GS1	1,3	0.0 (1)	0.0	13	654	6.14	Optimal	1,3	0.0 (1)	0.33	2	45	3.19	20
GS2	1,2	0.0	0.0	35	1227	10.29	Optimal	1,2	0.0	0.47	17	466	6.52	100
GS3	0,98	0.0	0.0	16	682	8.27	Optimal	0,97	0.0 (1)	0.45	4	87	5.26	20
GS4	4,7	0.0	0.0	35	1370	12.61	Optimal	4,7	0.0	0.44	20	618	9.35	20
GS5	1,2	0.0	0.0	20	1551	9.95	Optimal	1,2	0.0 (1)	0.14	1	46	3.46	20
GS6	13,3	0.0	0.0	83	3987	29.95	Optimal	13,2	0.0	0.66	24	932	11.12	120
GS7	3,5	0.2 (1)	0.0	29	1784	14.3	Optimal	3,4	0.2	0.55	4	100	5.93	20
GS8	-16,3	23 (2)	0.0	81	3965	31.39	Optimal	-16,4	23.0	0.57	35	1346	16.37	90
GM1	34,1	0.0	0.0	277	13413	98.12	Optimal	33,5	0.0	1.69	17	524	24.67	20
GM2	84,7	0.0	0.0	539	24162	259.02	Optimal	80,7	0.0	4.69	32	1249	33.13	20
GM3	24,6	0.0	0.0	287	12883	118.87	Optimal	24,4	0.0 (1)	0.92	48	1254	54.52	20
GM4	29,7	0.0	0.0	534	23582	204.42	Optimal	28,9	0.0 (1)	2.82	85	2954	70.33	20
GM5	45,3	0.0	0.0	624	24040	235.72	Optimal	44,5	0.0 (1)	1.77	19	615	38.55	20
GM6	144,8	0.0	0.0	1243	16982	2523.54	Optimal	138,8	0.0	4.14	37	1451	57.46	20
GM7	68,4	0.0	0.0	621	25674	267.28	Optimal	67,3	0.0 (1)	1.62	51	1483	66.47	20
GM8	128,3	0.0	0.0	1225	12103	2800.09	Optimal	121,8	0.0 (1)	5.12	98	4011	100.26	20

Scalability Analysis: Models Constraints

Dataset	Customers	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)
GS5	20	75	small (5)	few (50)	36	20	1.8 (0.8)
GS6	20	75	large (15)	few (50)	805	83	4 (3)
GS7	20	75	small (5)	many (100)	72	29	4 (2)
GS8	20	75	large (15)	many (100)	729	81	8 (6)
GM1	100	100	small (5)	few (300)	1144	277	17 (11)
GM2	100	100	small (5)	few (300)	5918	539	32 (26)
GM3	100	100	small (5)	many (800)	1232	287	48 (32)
GM4	100	100	large (15)	many (800)	5742	534	85 (69)
GM5	200	125	small (5)	few (300)	3138	624	19 (13)
GM6	200	125	large (15)	few (300)	18118	1243	37 (31)
GM7	200	125	small (5)	many (800)	3264	621	51 (35)
GM8	200	125	large (15)	many (800)	- (out of memory)	1225	98 (82)
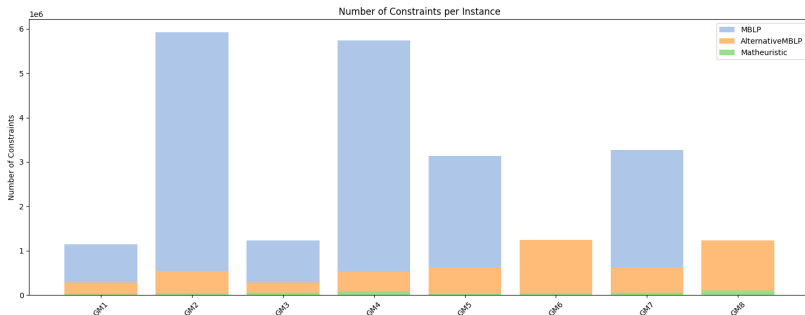
Scalability Analysis: CPU Time (sec) GM Datasets
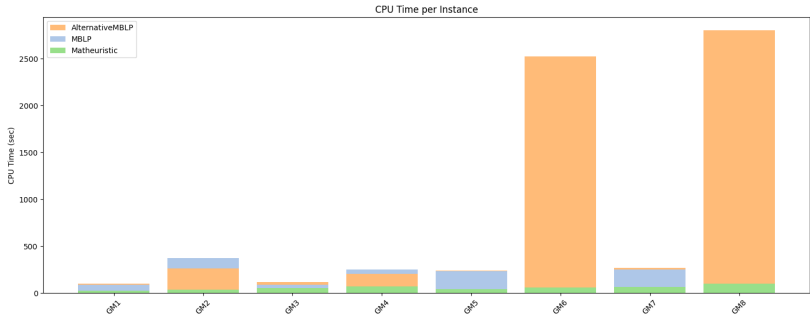
Memory Usage per Instance
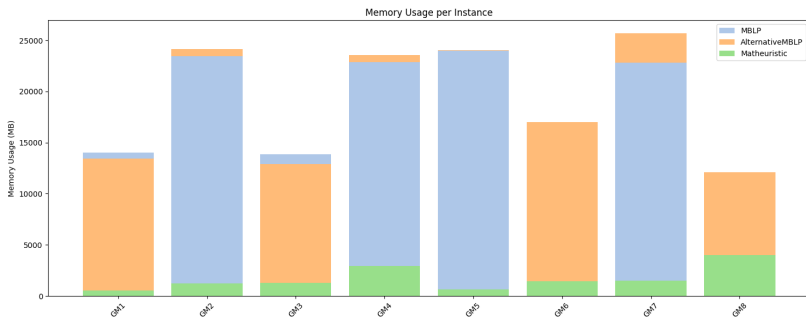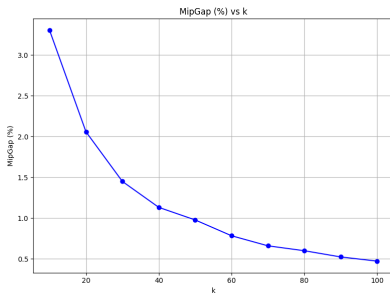
MIP Gap vs. k
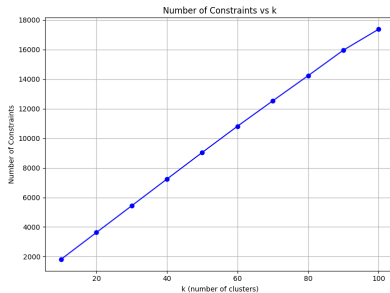
Number of Constraints vs. k

Matheuristic: Performance Overview

- **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

- Penalty costs from slack variables generally low

- Matheuristic **significantly faster** than the alternative MBLP

- Always stays within the company's time limit, even for very large real-world instances

- Solution quality is **more sensitive to eligibility fraction** than to the number of patterns

- **Scalability is excellent**: maintains high solution quality as instance size grows

We now analyze:

- **Impact of Step 5 of the preprocessing technique:**
  Removal of redundant subcliques at the group level and its
  effect on the performance of the matheuristic

- **Impact of the new modeling technique:**
  Group-level approach to conflict constraints proposed in the
  article and its influence on the performance of the matheuristic

Matheuristic without the New Modeling Technique

Benchmark version of the matheuristic:

- In this version, the matheuristic does not use the new modeling technique for handling conflict constraints

- 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:**

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

- Serves as a baseline to evaluate the impact of the new modeling technique

Matheuristic: new modeling impact

	MATHEURISTIC							MATHEURISTIC WITHOUT NEW MODELING TECHNIQUE						
	OFV (100k)	PENALTY (100K)	MIPGAP (%)	CONSTR. (1K)	MEMORY (MB)	CPU (SEC)	K	OFV (100k)	PENALTY (100K)	MIPGAP (%)	CONSTR. (1K)	MEMORY (MB)	CPU (SEC)	K
GS1	1,3	0.0 (1)	0.33	2	45	3.19	20	0,3	0.8	70.86	4	46	3.3	20
GS2	1,2	0.0	3.15	3	106	3.42	20	-0,6	1.1	105.2	17	114	3.96	20
GS3	0,97	0.0 (1)	0.45	4	87	5.26	20	0,8	0.0 (1)	8.42	9	92	5.5	20
GS4	4,6	0.0	2.13	7	225	6.46	20	3,7	0.0 (1)	19.96	36	244	7.21	20
GS5	1,2	0.0 (1)	0.14	1	46	3.46	20	1,2	0.0 (1)	5.51	2	47	3.71	20
GS6	13	0.0	2.26	4	169	4.81	20	10,8	0.5	18.65	40	195	6.22	20
GS7	3,4	0.2	0.55	4	100	5.93	20	3,1	0.3	10.91	8	104	6.83	20
GS8	-16,6	23.0	1.81	8	322	7.94	20	-18,2	24.9	11.66	72	364	9.92	20
GM1	33,5	0.0	1.69	17	524	24.67	20	26,2	3.3	22.96	69	562	31.55	20
GM2	80,7	0.0	4.69	32	1249	33.13	20	46,9	17.3	44.61	355	1451	47.96	20
GM3	24,4	0.0 (1)	0.92	48	1254	54.52	20	-15,1	35.3	161.16	199	1335	71.81	20
GM4	28,9	0.0 (1)	2.82	85	2954	70.33	20	-84,8	107.3	385.08	915	3486	109.52	20
GM5	44,5	0.0 (1)	1.77	19	615	38.55	20	36,8	1.3	18.72	94	664	51.94	20
GM6	138,8	0.0	4.14	37	1451	57.46	20	58,4	52.2	139.12	543	1770	82.84	20
GM7	67,3	0.0 (1)	1.62	51	1483	66.47	20	50,4	6.4	26.3	262	1600	102.68	20
GM8	121,8	0.0 (1)	5.12	98	4011	100.26	20	58,1	36.1	54.7	1446	4856	169.16	20

Matheuristic: new modeling impact

		Matheuristic	Matheuristic without step 5	Matheuristic without new modeling technique
Number of Constraints	∑ GS	34	294 (+764%)	188 (+452%)
	∑ GM	387	6507 (+1581%)	3883 (+903%)
Number of Conflict Constraints	∑ GS	22	282 (+1181%)	22
	∑ GM	299	6419 (+2046%)	299

- **Step 5** is critical for **reducing** the number of **conflict constraints**
- **Both** Step 5 of the preprocessing and the new modeling technique **drastically reduce the total number of constraints** in the heuristic

Conclusions

The study addresses a real-world planning problem from a telecommunications company: assigning customers to direct marketing activities while respecting complex business and customer-specific constraints. Existing methods ignore customer-specific constraints and are unsuitable for this problem.

A **matheuristic approach** is developed, which:

- Solves the problem first at a **group level**, then assigns individual customers iteratively

- Introduces **new modeling and preprocessing techniques** to handle customer-specific constraints efficiently

Conclusions

Results:

- The new modeling technique alongside the preprocessing step **drastically reduce the number of constraints** in the model, improving the matheuristic performance

- The matheuristic achieves an **average gap** of only **1.8%** from optimal solutions on generated instances

- Increasing the number of groups further reduces the gap, with minimal impact on runtime

- The method is adopted by the company, improving **sales** by **90%** and campaign **profitability** by around **300%**

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