# Algorithmic Methods for Mathematical Models – COURSE PROJECT –

Àlex Font Mercadié i Carles Matoses Gimenez

Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya

December 2024

# Table of Contents

# Problem Formal Statement

*Given*:

- The set of $D$ departments of the university indexed by $\{1, \ldots, D\}$.
- The set of $N$ faculty members indexed by $\{1, \ldots, N\}$. For each member $i$ its department $d_i$ is specified.
- The compatibility $m_{ij}$ between every pair of members $i$ and $j$.

*Find* the commission of members of the faculty subject to the following constraints:

- Each department $p$ has exactly $n_p$ members in the commission.
- There cannot be two members $i$ and $j$ in the commission with compatibility $m_{ij} = 0$.
- For each pair of members $i$ and $j$ in the commission with a compatibility lower than $0.15$ there must be a member $k$ in the commission with compatibility higher than $0.85$ with both members $i$ and $j$.

with the *objective* to maximize the average compatibility among all pairs of participants in the commission.

# ILP Formulation

| | |
|---|---|
| $N$ | Total number of faculty members, index $i$. |
| $D$ | Total number of faculty departments, index $p$. |
| $n_p$ | Number of members of department $p$ required in the commission. |
| $d_i$ | Department to which member $i$ belongs. |
| $m_{ij}$ | Compatibility between members $i$ and $j$. |

The following decision variables are also defined:

| | |
|---|---|
| $c_i$ | Binary variable representing if faculty member $i$ is selected to be part of the committee. |
| $x_{ij}$ | Binary variable representing if both faculty member $i$ and $j$ are selected to be part of the committee. |

# ILP Formulation

maximize

$$\sum_{i=1}^{N} \sum_{j=i+1}^{N} m_{ij} \cdot x_{ij}$$

subject to:

$$x_{ij} \leq c_i \qquad \forall i,j \in \{1,\ldots,N\} \tag{1}$$

$$x_{ij} \leq c_j \qquad \forall i,j \in \{1,\ldots,N\} \tag{2}$$

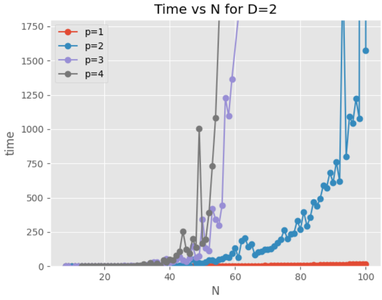
$$x_{ij} \geq c_i + c_j - 1 \qquad \forall i,j \in \{1,\ldots,N\} \tag{3}$$

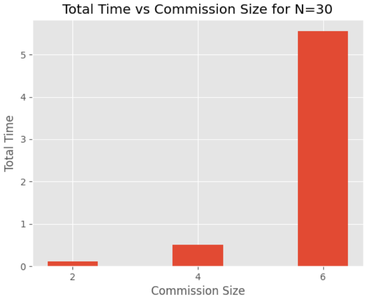
$$c_i + c_j \leq 1 \qquad \forall i,j \in \{1,\ldots,N\} \text{ s.t. } m_{ij} = 0 \tag{4}$$

$$1 - x_{ij} + \sum_{\substack{1 \leq k \leq N, \\ m_{ik}, m_{jk} > 0.85}} c_k \geq 1 \qquad \forall i,j \in \{1,\ldots,N\} \text{ s.t. } m_{ij} < 0.15 \tag{5}$$

$$\sum_{\substack{1 \leq i \leq N \\ d_i = p}} c_i = n_p \qquad \forall p \in \{1,\ldots,D\} \tag{6}$$

# ILP Formulation




| N | D | p | size | time |
|---|---|---|---|---|
| 30 | 2 | 4 | 8 | 3.47 |
| 30 | 4 | 2 | 8 | 0.56 |
| 54 | 2 | 4 | 8 | 1083.16 |
| 54 | 4 | 2 | 8 | 309.06 |

# Feasibility

**Algorithm 1:** isFeasible($C$)

**Input:** A set $C$ of faculty members.

**foreach** *department* $p \in \{1, \ldots, D\}$ **do**
    **if** *number of members from department* $p$ *in* $C \neq n_p$ **then**
        **return false**

**foreach** *pair of members* $i, j \in C$ **do**
    **if** $m_{ij} = 0$ **then**
        **return false**

**foreach** *pair of members* $i, j \in C$ **do**
    **if** $m_{ij} < 0.15$ **then**
        **if** *there is no member* $k \in C$ *s.t.* $m_{ik} > 0.85$ *and* $m_{jk} > 0.85$ **then**
            **return false**

# Committee members' weight

The weight of a faculty member $i$ with respect to a set $C$ of faculty members.

$$w(i, C) = \sum_{j \in C} m_{ij} \tag{7}$$

For the committee $C$, the objective function can be computed as the average compatibility among all pairs of members in the committee.

$$\text{average\_compatibility}(C) = \frac{1}{|C|} \sum_{i \in C} \sum_{j \in C,\ i<j} m_{ij} \tag{8}$$

# Committee members' weight (our approach)

$$w(i, C) = w_1(i, C) + w_2(i, C) \tag{9}$$

The following formula takes into account the relationship of the current committee members and the candidate $i$:

$$w_1(i, C) = \sum_{j \in C} \begin{cases} 2m_{ij}, & \text{if } 0.15 \leq m_{ij} \leq 0.85 \text{ or intermediate}(i, j, C), \\ -2, & \text{if } m_{ij} < 0.15 \text{ and not intermediate}(i, j, C) \end{cases} \tag{10}$$

For further refinement, the algorithm will also take into account the future candidates members not in the committee:

$$w_2(i, C) = \sum_{j \notin C} \begin{cases} m_{ij}, & \text{if } 0.15 \leq m_{ij} \text{ or intermediate}(i, j, C) \text{ and isValid}(j, C \cup \{i\}), \\ -1, & \text{if } m_{ij} < 0.15\text{and not intermediate}(i, j, C), \text{ and isValid}(j, C \cup \{i\}). \end{cases} \tag{11}$$

# Greedy Constructive Algorithm: Variables

$N$: Total number of members
$D$: Total number of departments
$n$: Array where $n_p$ is the required members capacity of department $p$ $(1 \leq p \leq D)$
$d$: Array where $d_i$ is the department of member $i$ $(1 \leq i \leq N)$
$m$: Compatibility matrix where $m_{ij}$ represents compatibility between members $i$ and $j$

Initialize $C \leftarrow \emptyset$
Initialize $members \leftarrow \{1, \ldots, N\}$

# Greedy Constructive Algorithm: Loop

**while** $C$ *not a solution* **do**
    $feasible\_members \leftarrow \{\}$
    $members\_weights \leftarrow \{\}$
    **foreach** $member \notin C$ **do**
        **if** *isFeasible(member, C)* **then**
            $feasible\_members \leftarrow feasible\_members \cup \{member\}$
            $members\_weights[member] \leftarrow w(member, C)$
    **if** $|feasible\_members| > 0$ **then**
        $best\_member \leftarrow \arg\max_{m \in feasible\_members} members\_weights[m]$
        $C \leftarrow C \cup \{best\_member\}$

# Greedy Constructive Algorithm: Feasible

To check if the solution is feasible, we can check the not computed constraints.

**foreach** *pair of members* $(i, j) \in C \times C$ **do**
  **if** $m_{ij} < 0.15$ **then**
    **if** *there does not exist a* $k \in C$ *such that* $m_{ik} > 0.85$ *and* $m_{jk} > 0.85$ **then**
      Print "Relaxed solution is not valid for the original problem"
      **return** $C$ *// Relaxed solution to be improved later*

# Local Search

```
C ← current_solution
Comp ← average_compatibility(C)
foreach i ∈ {1, ..., N} do
    if i ∈ C then
        C_neighbor ← C
        C_neighbor ← remove(C_neighbor, i)
        foreach j ∈ {1, ..., N} do
            if j ∉ C_neighbor then
                C_neighbor ← add(C_neighbor, j)
                if isFeasible(C_neighbor) then
                    Comp_neighbor ← average_compatibility(C_neighbor)
                    if Comp_neighbor > Comp then
                        Comp ← Comp_neighbor
                        C ← C_neighbor
```

# GRASP

C $\leftarrow \emptyset$
Candidates $\leftarrow \{1, \ldots, N\}$
**while** *C not a solution* **do**
    W $\leftarrow [0, 0, \ldots, 0]$ of size $N$
    **foreach** $n \in$ *Candidates* **do**
        $W_n \leftarrow w(n, C)$
    $W_{min} \leftarrow \min\{W_n \mid n \in \text{Candidates}\}$
    $W_{max} \leftarrow \max\{W_n \mid n \in \text{Candidates}\}$
    $RCL_{max} \leftarrow \{n \in \text{Candidates} \mid W_n \geq W_{max} - \alpha(W_{max} - W_{min})\}$
    Select $n \in RCL$ at random
    $C \leftarrow C \cup \{n\}$
    Candidates $\leftarrow$ Candidates $\setminus \{n\}$

# GRASP

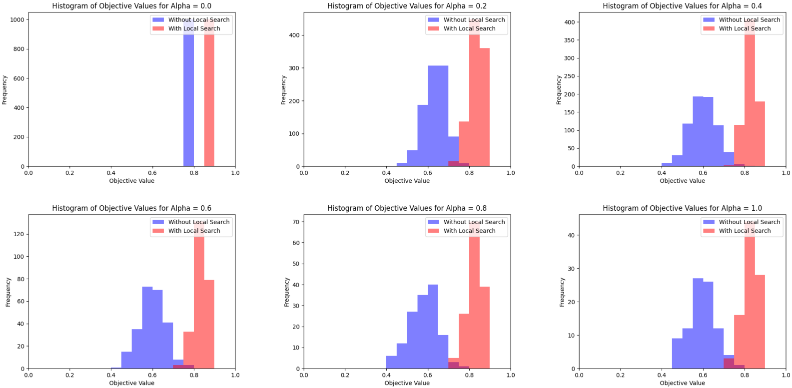
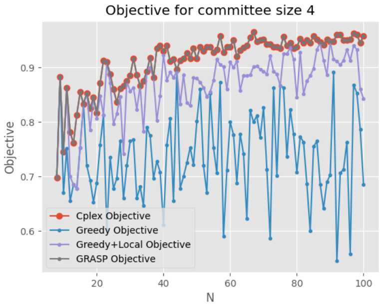
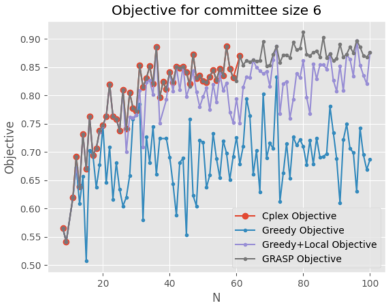



Figure: Distribution of construction phase solution values as a function of the RCL parameter $\alpha$ ($1000$ repetitions were recorded for each value of $\alpha$)

# GRASP

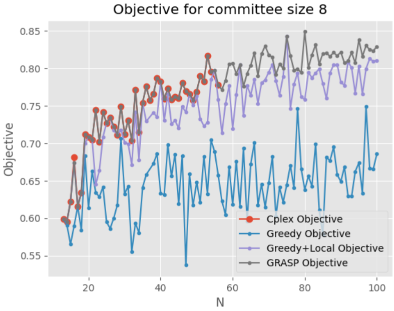


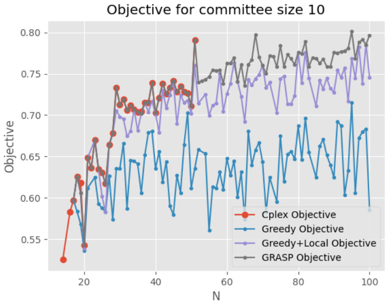
Objective for committee size 4
Objective
0.9
0.8
0.7
0.6
20
40
60
80
100
N
Cplex Objective
Greedy Objective
Greedy+Local Objective
GRASP Objective

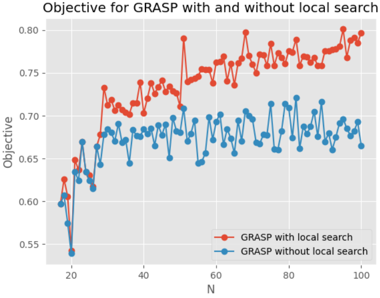
Objective for committee size 6
Objective
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
20
40
60
80
100
N
Cplex Objective
Greedy Objective
Greedy+Local Objective
GRASP Objective

# GRASP

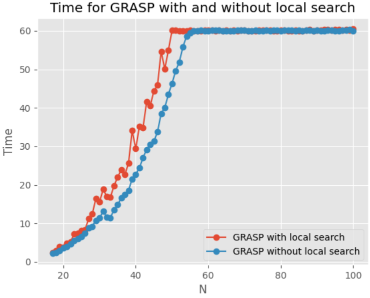

Objective for committee size 8
Objective
N
0.85
0.80
0.75
0.70
0.65
0.60
0.55
20
40
60
80
100
Cplex Objective
Greedy Objective
Greedy+Local Objective
GRASP Objective


Objective for committee size 10
Objective
N
0.80
0.75
0.70
0.65
0.60
0.55
20
40
60
80
100
Cplex Objective
Greedy Objective
Greedy+Local Objective
GRASP Objective

# GRASP



Objective for GRASP with and without local search
Objective
N
0.80
0.75
0.70
0.65
0.60
0.55
20
40
60
80
100
GRASP with local search
GRASP without local search

Time for GRASP with and without local search
Time
N
60
50
40
30
20
10
0
20
40
60
80
100
GRASP with local search
GRASP without local search