Multiobjective metaheuristic in three stages to plan a speed meeting
Part 1: Peek

Delavernhe Florian, Legru Guillaume, Lucas Flavien, Robbes Alexis

[a] *Université de Nantes – Faculty of Sciences*
2 Rue de la Houssinière BP 92208 F44322 Nantes Cedex 03 – France

---

Abstract

This paper discuss on the 3-phases method (GRASP, Genetic Algorithm and local search), applied to a speed meeting problem. More precisely, this paper specify the efficiency of an initial population created by the GRASP Algorithm.

Keywords: Multi-objective Optimisation, Heuristic, 3-phases method, Time-tabling Problem.

---

*Corresponding authors
Email addresses: `florian.delavernhe@etu.univ-nantes.fr` (Delavernhe Florian), `guillaume.legrus@etu.univ-nantes.fr` (Legru Guillaume), `flavien.lucas@etu.univ-nantes.fr` (Lucas Flavien), `alexis.robbes@etu.univ-nantes.fr` (Robbes Alexis)
URL: `https://github.com/Shakit/MOMH_Project_2016` (Legru Guillaume)

Contents

1. Introduction

1.1. Context

The time-tabling problem is a satisfaction problem, but in some case it is not necessary to respect all the constraints. In this type of case we can create an optimization problem. The objective function is to satisfy the maximum number of constraints.

. Our problem is to plan a speed meeting between industrialists and researchers. For each time slots the number of meeting is restricted by the number of available rooms. The optimization part lies in maximize the satisfaction of industrialists and researchers.

. Define the satisfaction is the multi-objective aspect of the problem. It is depend of whose have been met and the tiredness due to the move between each meeting.

1.2. Notations and definitions

$$\text{Datas:} \begin{cases} I & \text{set of the industrialists} \\ R & \text{set of the researchers} \\ S & \text{set of the rooms} \\ H & \text{set of time slots} \\ H' & \text{set of time slots without the first one} \\ p_{ir} & \text{is the preference value of } i \text{ in meeting } r \quad \forall (i,r) \in I \times R \\ p'_{ri} & \text{is the preference value of } r \text{ in meeting } i \quad \forall (r,i) \in R \times I \\ d_{st} & \text{is the distance between the room } s \text{ and } t \quad \forall (s,t) \in S^2 \end{cases}$$

$$\text{Variables:} \begin{cases} x_{ir}^{sh} & \text{is 1 if } i \text{ m} \\ y_i^h & \text{is the dist} \\ z_r^h & \text{is the dist} \\ p_{min} & \text{is the min} \\ d_{max} & \text{is the max} \end{cases}$$

$$
\begin{array}{lll}
1) & \max & \sum_{i\in I}\sum_{r\in R}\sum_{s\in S}\sum_{h\in H}(p_{ir}+p'_{ri}).x_{ir}^{sh} \\
2) & \max & p_{min} \\
3) & \min & \sum_{h\in H'}(\sum_{i\in I} y_i^h + \sum_{r\in R} z_r^h) \\
4) & \min & d_{max}
\end{array}
$$

$s.c$

$$
\begin{array}{lll}
(1) & \sum_{r\in R}\sum_{s\in S} x_{ir}^{sh} \leq 1 & \forall(i,h)\in I\times H \\
(2) & \sum_{i\in I}\sum_{s\in S} x_{ir}^{sh} \leq 1 & \forall(r,h)\in R\times H \\
(3) & \sum_{i\in I}\sum_{r\in R} x_{ir}^{sh} \leq 1 & \forall(s,h)\in S\times H \\
(4) & \sum_{h\in H}\sum_{s\in S} x_{ir}^{sh} \leq 1 & \forall(i,r)\in I\times R \\
(5) & d_{st}.(\sum_{r\in R}(x_{ir}^{sh-1}+x_{ir}^{th})-1) \leq y_i^h & \forall(s,t,i,h)\in S^2\times I\times H' \\
(6) & d_{st}.(\sum_{i\in I}(x_{ir}^{sh-1}+x_{ir}^{th})-1) \leq z_r^h & \forall(s,t,r,h)\in S^2\times R\times H' \\
(7) & p_{min} \leq \sum_{h\in H}\sum_{s\in S}\sum_{r\in R} p_{ir}.x_{ir}^{sh} & \forall i\in I \\
(8) & p_{min} \leq \sum_{h\in H}\sum_{s\in S}\sum_{i\in I} p'_{ri}.x_{ir}^{sh} & \forall r\in R \\
(9) & d_{max} \geq y_i^h & \forall(i,h)\in I\times H' \\
(10) & d_{max} \geq z_r^h & \forall(r,h)\in R\times H' \\
 & x_{ir}^{sh}\in\{0,1\} & \forall(i,r,s,h)\in I\times R\times S\times H \\
 & y_i^h\in\mathbb{R}_+ & \forall(i,h)\in I\times I\times H' \\
 & z_r^h\in\mathbb{R}_+ & \forall(r,h)\in R\times H'
\end{array}
$$

Objectives functions: $\begin{cases} \text{1) maximize the total preference} \\ \text{2) maximize the minimum preference} \\ \text{3) minimize the total distance traveled} \\ \text{4) minimize the maximum distance traveled} \end{cases}$

Constraints:

(1): each industrialist have at most one meeting in one room by time slots

(2): each researcher have at most one meeting in one room by time slots

(3): at most one meeting for each room by time slots

(4): industrialist meet at most once a researcher

(5): define the distance traveled by each industrialists between two time slots

(6): define the distance traveled by each researchers between two time slots

(7) and (8): give the minimum preference value in the assignment

(9) and (10): give the maximum distance traveled in the assignment

. In this part we will work first on a bi-objectives version:

$$
(P_{bi})\left\{\begin{array}{lll}
 & \max \quad \sum\limits_{i\in I}\sum\limits_{r\in R}\sum\limits_{s\in S}\sum\limits_{h\in H}(p_{ir}+p'_{ri}).x_{ir}^{sh} & \\
 & \min \quad \sum\limits_{h\in H'}(\sum\limits_{i\in I}y_i^h+\sum\limits_{r\in R}z_r^h) & \\
s.c & & \\
 & (1) \quad \sum\limits_{r\in R}\sum\limits_{s\in S}x_{ir}^{sh}\leq 1 & \forall(i,h)\in I\times H \\
 & (2) \quad \sum\limits_{i\in I}\sum\limits_{s\in S}x_{ir}^{sh}\leq 1 & \forall(r,h)\in R\times H \\
 & (3) \quad \sum\limits_{i\in I}\sum\limits_{r\in R}x_{ir}^{sh}\leq 1 & \forall(s,h)\in S\times H \\
 & (4) \quad \sum\limits_{h\in H}\sum\limits_{s\in S}x_{ir}^{sh}\leq 1 & \forall(i,r)\in I\times R \\
 & (5) \quad d_{st}.(\sum\limits_{r\in R}(x_{ir}^{sh-1}+x_{ir}^{th})-1)\leq y_i^h & \forall(s,t,i,h)\in S^2\times I\times H' \\
 & (6) \quad d_{st}.(\sum\limits_{i\in I}(x_{ir}^{sh-1}+x_{ir}^{th})-1)\leq z_r^h & \forall(s,t,r,h)\in S^2\times R\times H' \\
 & \quad\quad\; x_{ir}^{sh}\in\{0,1\} & \forall(i,r,s,h)\in I\times R\times S\times H \\
 & \quad\quad\; y_i^h\in\mathbb{R}_+ & \forall(i,h)\in I\times I\times H' \\
 & \quad\quad\; z_r^h\in\mathbb{R}_+ & \forall(r,h)\in R\times H'
\end{array}\right.
$$

2. Instances

2.1. Definition sets

. Preferences values:
We choose that the value of each preference is a rating between 0 and 100. As in the model, we keep the difference between the industrialists preferences and the researcher preferences.

. Distances values:
We choose to work with euclidean distances (rounded to the nearest integer), that means the distance doesn't depend of the sens and the distance between two rooms is lesser or equal that the distance of passing by a third one.

. Numder of industrialists and researcher:
To break some symmetries we choose that the number of researchers would be greater or equal to the number of industrialists

. Number of rooms:
As it is useless to have more room than the maximum of meeting possible by time slot: $\min\{I, R\}$ We decide to limit the number of room at this bound.

. Number of time slots:
This parameters is maybe the worst for a resolution because that increases the number of permutation for a set of meeting. These permutations play an essential role for the second objective (minimize the distance traveled).

3. Difficulties of an exact resolution

3.1. NP-Complete

If we focus on maximize the total preference the problem become a set packing problem:

$$(P_1)\left\{\begin{array}{lll} & \max \quad \sum_{i\in I}\sum_{r\in R}\sum_{s\in S}\sum_{h\in H}(p_{ir}+p'_{ri}).x_{ir}^{sh} & \\ s.c & & \\ & (1) \quad \sum_{r\in R}\sum_{s\in S} x_{ir}^{sh} \leq 1 & \forall (i,h)\in I\times H \\ & (2) \quad \sum_{i\in I}\sum_{s\in S} x_{ir}^{sh} \leq 1 & \forall (r,h)\in R\times H \\ & (3) \quad \sum_{i\in I}\sum_{r\in R} x_{ir}^{sh} \leq 1 & \forall (s,h)\in S\times H \\ & (4) \quad \sum_{h\in H}\sum_{s\in S} x_{ir}^{sh} \leq 1 & \forall (i,r)\in I\times R \\ & \quad\quad x_{ir}^{sh}\in\{0,1\} & \forall (i,r,s,h)\in I\times R\times S\times H \end{array}\right.$$

As the speed meeting problem is a special case of the set packing problem which is a NP-Complete problem, we can conclude that the problem is at least NP-Hard.

3.2. Enumeration

It is

3.3. 2 phases method

Using the modeling language GNU MathProg and the GLPK solver we try to set up a 2 phases method. Even compute the Lexicographic optimal solution could be a hard task. The major problem is with the growth of the number of time slots.

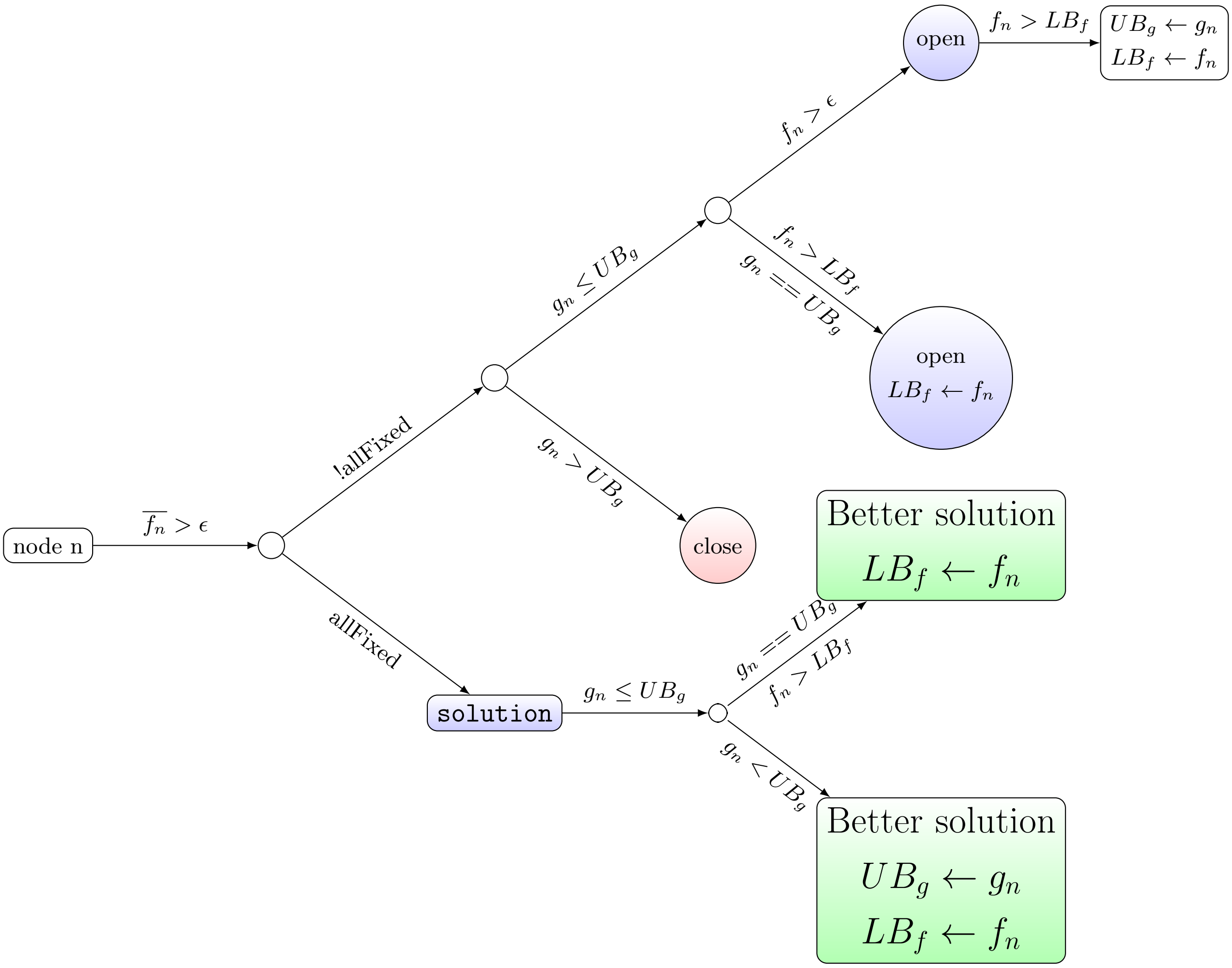
3.4. ϵ-constrain method

$$(P_\epsilon)\left\{\begin{array}{lll} & \min \quad \sum_{h\in H'}(\sum_{i\in I} y_i^h + \sum_{r\in R} z_r^h) & \\ s.c & & \\ & (1) \quad \sum_{r\in R}\sum_{s\in S} x_{ir}^{sh} \leq 1 & \forall (i,h)\in I\times H \\ & (2) \quad \sum_{i\in I}\sum_{s\in S} x_{ir}^{sh} \leq 1 & \forall (r,h)\in R\times H \\ & (3) \quad \sum_{i\in I}\sum_{r\in R} x_{ir}^{sh} \leq 1 & \forall (s,h)\in S\times H \\ & (4) \quad \sum_{h\in H}\sum_{s\in S} x_{ir}^{sh} \leq 1 & \forall (i,r)\in I\times R \\ & (5) \quad d_{st}.(\sum_{r\in R}(x_{ir}^{sh-1}+x_{ir}^{th})-1) \leq y_i^h & \forall (s,t,i,h)\in S^2\times I\times H' \\ & (6) \quad d_{st}.(\sum_{i\in I}(x_{ir}^{sh-1}+x_{ir}^{th})-1) \leq z_r^h & \forall (s,t,r,h)\in S^2\times R\times H' \\ & (\epsilon) \quad \sum_{i\in I}\sum_{r\in R}\sum_{s\in S}\sum_{h\in H}(p_{ir}+p'_{ri}).x_{ir}^{sh} > \epsilon & \\ & \quad\quad x_{ir}^{sh}\in\{0,1\} & \forall (i,r,s,h)\in I\times R\times S\times H \\ & \quad\quad y_i^h\in\mathbb{R}_+ & \forall (i,h)\in I\times I\times H' \\ & \quad\quad z_r^h\in\mathbb{R}_+ & \forall (r,h)\in R\times H' \end{array}\right.$$

. Bi-objectives Adaptive Branch and bound:

```
Result: Return all efficients solutions
ε ← 0;
Open-nodes ← ∅;
Close-nodes ← ∅;
Solutions ← ∅;
Efficients ← ∅;
add the full unfix node in Open-nodes;
while ε < value of an optimal solution of (P1) do
    lower bound of f ← 0;
    upper bound of g ← ∞;
    search a solution from Solutions and update bounds;
    while Open-nodes ≠ ∅ do
        find the node from Open-nodes with the maximum fixed variables;
        Child1 ← fix the first unfixed variable to 1;
        Child0 ← fix the first unfixed variable to 0;
        remove the parent node from Open-nodes;
        add Child1 and Child0 to the respect collection;
        update the bounds;
    end
    ε ← lower bound of f;
    fill Open-nodes with Close-nodes;
    clear Close-nodes;
end
```

Algorithm 1: Bi-objectives Adaptive Branch and bound



3.5. Lagrangian relaxation

4. Elaboration de la population initiale

4.1. Hypothèses et notations

- aucune solution identique
- solutions équivalentes autorisées

- taille de la population : $nInd$
- % solutions aléatoires (simplement réalisables) : $propIndRnd$
- % solutions construites (sous-optimales réalisables) : $propIndConstruits$

- Ensemble de solutions construites : $setSolB$
- Ensemble de solutions aléatoires : $setSolR$
- population d'individus utilisée par l'algorithme : $Y_A = \text{setSolB} \cup \text{setSolR}$

- nombre de direction d investiguées : $nDir$

- à une direction d correspond un vecteur poids λ, avec $\lambda \in \Lambda$ l'ensemble des vecteurs poids

4.2. Initial population development

Une population initiale Y_A composée de $nInd$ individus peut résulter de l'union de deux ensembles de solutions :

- des solutions aléatoires; $setSolR$

- des solutions construites; $setSolB$

Elle peut également considérer des solutions réalisables et non-réalisables. Dans la suite, seul le cas de solutions réalisables est abordé.

Elle peut se limiter à un ensemble de points potentiellement non-dominés Y_{PN} ou maintenir également des points dominés Y_D dans la population.

Cette section liste un ensemble de procédures envisageables pour élaborer ces deux ensembles de solutions.

SUPPRIMER CE QUI EST HORS CONTEXTE VOUS CONCERNANT

4.2.1. Solutions construites

- aucune information
  - l'ensemble des solutions construites est donc vide, la population sera composée que de solutions aléatoires.
- information exacte
  - la solution (lex-)optimale pour chaque objectif
  - une partie de solutions efficaces (obtenu par application d'une méthode de type somme pondérée, ou encore ϵ-contrainte)
  - Tout ou partie des solutions supportées
- information approchée
  - une solution par direction d
    - usage d'un branch-and-bound limité
    - exploitation d'une solution "experte"
    - usage d'un algorithme de construction de type glouton
    - usage d'une (meta)heuristique
    - application d'une relaxation suivi d'une réparation
  - plusieurs solutions par direction d
    - usage de la phase de construction de la métaheuristique GRASP

4.2.2. Solutions aléatoires

- utilisant aucune information

    - solutions totalement aléatoires

- dérivées du premier ensemble de solutions construites :

    - solutions obtenues par mutation

    - solution obtenues par path-relinking, d'un scatter search

    - solutions obtenues par recherche locale

4.3. Algorithme GRASP comme méthode de construction

4.3.1. Rappel du principe

- phase de construction de $nIterGrasp$ solutions pour une valeur α fixe pré-établie

- pas de phase d'amélioration

4.3.2. Paramètres

- compromis glouton-aléatoire (α)

- nombre d'itération ($nIterGrasp$)

5. Experimentations

5.1. Sac-à-dos bi-dimensionnel ($m = 2$) en variables 0-1 avec $p \in \{2, 3\}$ objectifs (pO2DKP) comme problème support à l'étude

5.2. Directions pour la construction de seeding solutions

5.3. Procédure de construction de solutions GRASP

- utilité d'une variable : ???

- fonction de valeur utilisée pour la RCL???

5.4. Procédure de réparation 2DKP

- principe à expliciter!!!

- appliqué lors de l'élaboration d'une solution aléatoire

- appliqué après application d'un opérateur évolutionnaire (crossover)

5.5. Protocole :

5.5.1. Instances

5.5.2. Paramètres

. nombre d'individus : $nInd$=???

. pourcentage d'individus construits : $propIndConstruits = 50\%$

. pourcentage d'individus aléatoires : $propIndRnd = 1 - propIndGrasp$

. nombre de direction d investiguées : $nDir = nInd/10$

. nombre d'itération par direction : $nIterGrasp = nInd/(2 * nDir)$

. compromis glouton-aleatoire : α =??

5.5.3. Machine

???

5.5.4. Mesure de la qualité des approximations obtenues

Connaissant les points non-dominés exacts pour les instances utilisées, 3 mesures s'invitent naturellement :

. M_1 : nombre de points non-dominés exacts trouvés

. S-metric : hypervolume donné par l'approximation

. CPUt : temps de calcul écoulé

5.6. Résultats

6. Bilan

Conclusions, perspectives

Acknowledgment

Remerciements à ... pour ... .