



Planification quotidienne d'une équipe mobile



Cléa Chataigner, Tom Collignon, Antoine de Bouillé, Olivia Moyal, Victor Neyt

Introduction



Premier modèle exact simplifié



Deuxième modèle exact étendu



Métaheuristique



Conclusion





CentraleSupélec

Introduction



DecisionBrain

- Start-up ;
- Solutions d'optimisation basées sur l'IA ;
- Aide à la décision.

Notre client

- TuttoBene ;
- Logiciel d'aide à la décision ;
- Assister la planification de l'équipe mobile de techniciens.

Objectif

Créer les routes quotidiennes de **500 techniciens**, devant réaliser **10 000 tâches**.



CentraleSupélec

Premier modèle exact simplifié



Modélisation

Formalisation du Vehicle Routing Problem with Time Windows (VRPTW)

Variables de décision

- Variables de flux :

$$x_{ijk} = \begin{cases} 1 & \text{si l'arc } (i, j) \text{ est utilisé par l'employé } k \\ 0 & \text{sinon} \end{cases}$$

- Dépôt représenté par les nœuds 0 et n+1 ;
- Variables de temps w_i : heure à laquelle la tâche i commence.

Modélisation

Objectif

Minimiser les coûts de transport

$$\text{Minimize } \sum_{k=1}^{n_o} \sum_{i=0}^{n_t} \sum_{\substack{j=1 \\ j \neq i}}^{n_t+1} c_{ij} x_{ijk}$$

Contraintes

- Retour au dépôt ;
- Respect du débit ;
- Visite unique d'une tâche ;
- Respect des heures de travail ;
- Respect des niveaux de compétences ;
- Durée d'une tâche et transport ;
- Respect des horaires d'ouverture d'une tâche.

Premier modèle exact simplifié



CentraleSupélec

Modélisation

Objectif

Minimiser les coûts de transport

$$\text{Minimize } \sum_{k=1}^{n_o} \sum_{i=0}^{n_t} \sum_{\substack{j=1 \\ j \neq i}}^{n_t+1} c_{ij} x_{ijk}$$

Contraintes

Exemple des contraintes horaires des employés

$$\forall k \in K, i \in \{1, \dots, nt\}$$

$$(d_k + c_{0i} - wi) \leq \left(1 - \sum_{\substack{j=0 \\ j \neq i}}^{n_t} x_{jik} \right) * H$$

Premier modèle exact simplifié



CentraleSupélec

Analyse des résultats

Instances	Temps d'exécution (sec)	Nombre d'employés	Nombre de tâches
GuineaGolf	0,15	1	9
Bordeaux	3	2	10
Poland	259,04	2	19
Italy	771,71	3	21
Finland	> 20 000	4	25

Temps d'exécution de l'algorithme sur les instances

- Trop de temps d'exécution sur l'instance Finlande ;
- Nombre de contraintes en $\mathcal{O}(n_t^2 + Knt)$.

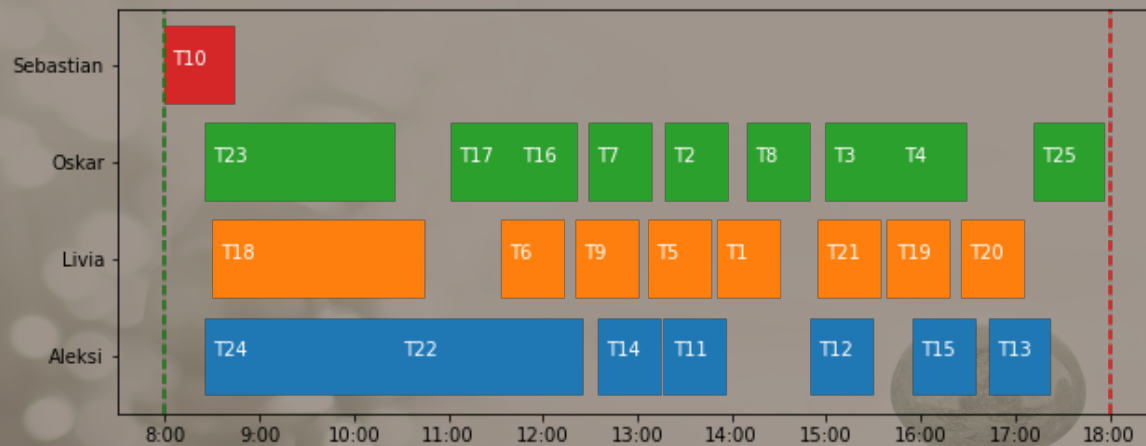
Premier modèle exact simplifié



CentraleSupélec

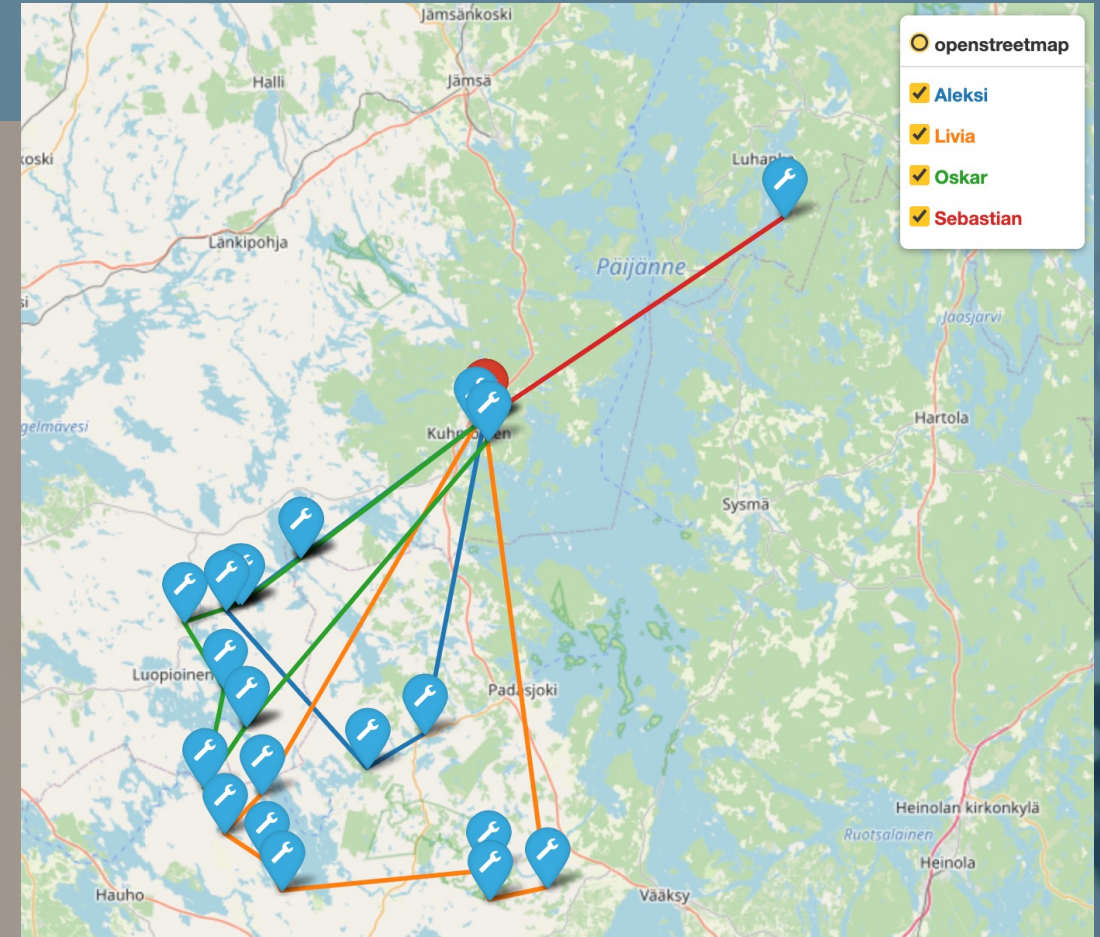
Analyse des résultats

Exemple de la Finlande



Emploi du temps

Écart à l'optimalité : 0.485901



Résultat cartographique



CentraleSupélec

Deuxième modèle exact étendu



Modélisation

Variables de décision

- Indisponibilités représentées comme des tâches (nœuds du graphe) ;
- Pour chaque employé, ensemble N_k : ensemble des tâches, sa maison H_k et les lieux de ses indisponibilités ;
- Mêmes variables de flux x_{ijk} ;
- Mêmes variables de temps w_i ;
- Variables p_k : pour chaque employé, heure de début de sa pause déjeuner.

Deuxième modèle exact étendu



CentraleSupélec

Modélisation

Objectif

Maximiser la durée totale des tâches réalisées...

...à coût opérationnel minimum.

$$\text{Maximize } \sum_{k \in K} \sum_{i \in T} \sum_{\substack{j \in N_k \\ j \neq i}} s_i x_{ijk}$$

$$\text{Minimize } \sum_{k \in K} \sum_{i \in N_k} \sum_{\substack{j \in N_k \\ j \neq i}} c_{ij} x_{ijk}$$

$$\text{Maximize } 0.8 * \sum_{k \in K} \sum_{i \in T} \sum_{\substack{j \in N_k \\ j \neq i}} s_i x_{ijk} - 0.08 * \sum_{k \in K} \sum_{i \in N_k} \sum_{\substack{j \in N_k \\ j \neq i}} c_{ij} x_{ijk}$$

Modélisation

Contraintes

- Respect du débit ;
- Visite unique de chaque nœud ;
- Visite obligatoire d'une indisponibilité (considérée comme une tâche) ;
- Respect des niveaux de compétences des agents ;
- Respect des heures de travail des employés ;
- Respect de la pause déjeuner ;
- Durée d'une « tâche » (vraie tâche et indisponibilité) et transport ;
- Respect des horaires d'ouverture d'une « tâche ».

Modélisation

Contraintes

Exemple de la pause déjeuner

$$(w_i + s_i - p_k) \leq (2 - \sum_{\substack{j \in N_k \\ j \neq i}} x_{jik} - \gamma_{i,k}) * H \quad \forall k \in K, i \in T \cup U$$

Analyse des résultats

Instances	Temps d'exécution (sec)	Nb d'employés	Indisponibilités (employés)	Nb de tâches	Indisponibilités (tâches)
Bordeaux	129	2	1	10	1
Poland	> 3 600	2	1	19	1
Australia	13	3	1	16	6
Spain	> 3 600	3	2	27	5
Austria	> 3 600	5	3	31	5

Temps d'exécution de l'algorithme sur les instances

- Trop de temps d'exécution sur les instances Pologne, Espagne, Autriche ;
- Nombre de contraintes en $\mathcal{O}((K + ni)n_t + (K + ni)\text{card}(U) + (n_t + \text{card}(U))^2)$.

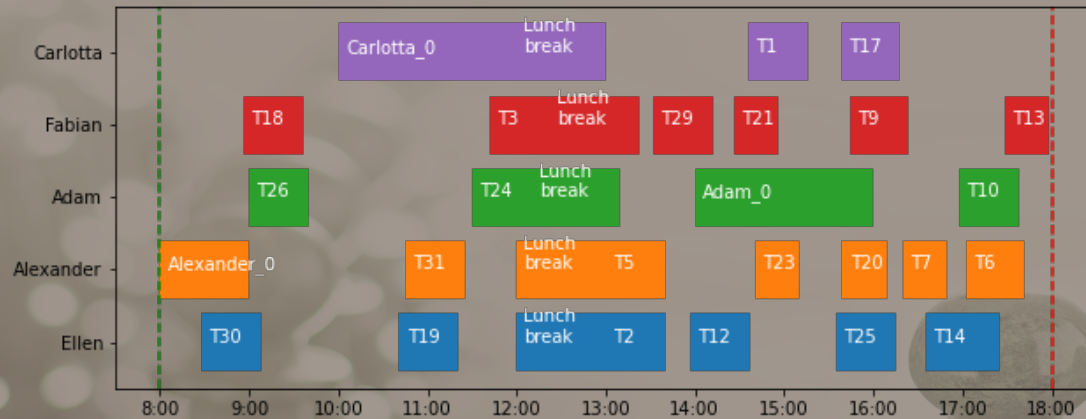
Deuxième modèle exact étendu



CentraleSupélec

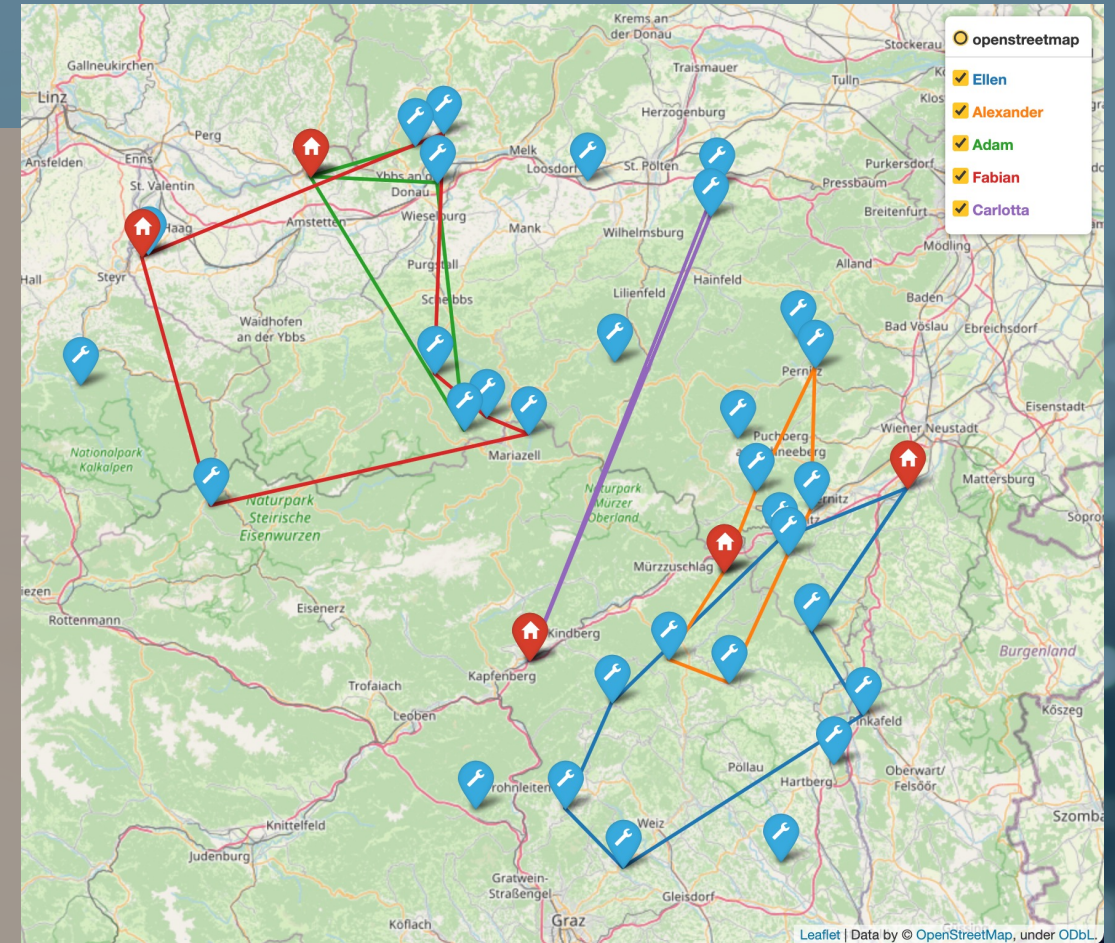
Analyse des résultats

Exemple de l'Autriche



Emploi du temps

Écart à l'optimalité : 0.394834



Résultat cartographique



CentraleSupélec

Métaheuristique



Modélisation

Initialisation de T , T_f et α

Initialisation d'un planning d'affectation

Tant que $T > T_f$:

- choix aléatoires d'une tâche à réaliser et d'un agent à affecter à la tâche ;
- calcul de l'ensemble des voisins possibles ;
- choix aléatoire d'un voisin parmi ceux possibles et calcul de sa fonction coût E_{voisin} .
- calcul $\Delta E = E_{\text{planning}} - E_{\text{voisin}}$;
- choix du nouveau planning :
 - si $\Delta E \geq 0$, alors le voisin choisi devient le planning
 - sinon, le voisin choisi devient le planning avec une probabilité $e^{-\Delta E/T}$
- calcul $T = T - \alpha$

Pseudo code

Pour toute tache :

On supprime la tache de l'agent auquel elle appartient

Pour tout agent :

On vérifie pour chacun de ses créneaux de disponibilités si la tâche est effectuable dans ce dernier

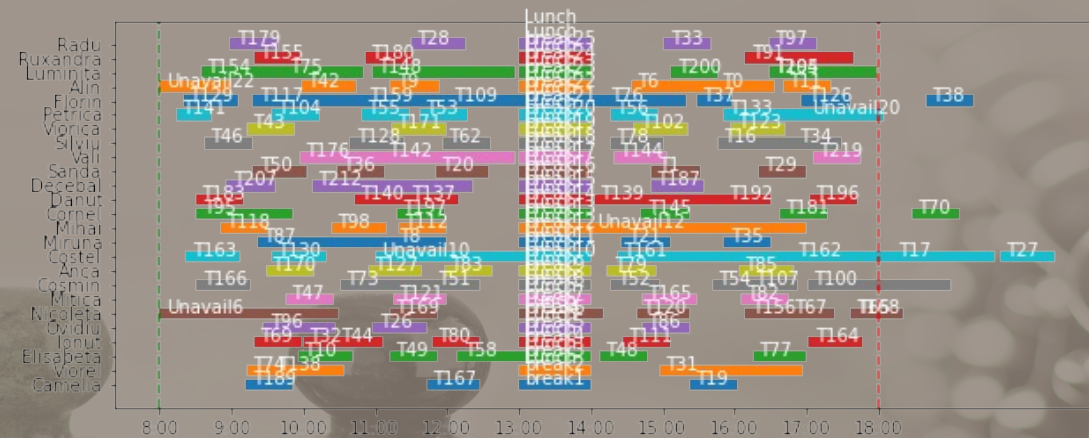
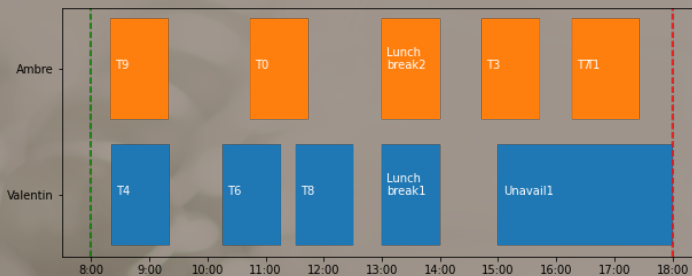
Pour chaque créneau où c'est le cas, la situation actualisée est le nouveau voisin

Chaque agent possède un emploi du temps sous forme de liste (en attribut). On peut directement en tirer ses créneaux de disponibilité et y appliquer les différentes contraintes.

Des résultats peu concluants...



CentraleSupélec





CentraleSupélec

Conclusion



Comparaison de nos résultats



CentraleSupélec

Instance	Bordeaux	Austria	Poland	Spain	Australie
Exact	129	> 3 600	> 3 600	> 3 600	13
Méta-heuristique	13	138	71	128	40

Comparaison des temps d'exécution en secondes,
pour 10 000 itérations dans la métaheuristique

- Corriger la métaheuristique
- Utiliser des échanges pour trouver les voisins
- Pouvoir automatiquement "glisser temporellement les tâches"
- Utiliser un algorithme glouton pour gagner encore du temps
- Mettre en place la méthode de l'évolution différentielle