

ST7-OSTP - Planification quotidienne d'une équipe mobile (DecisionBrain)

Responsables :

- Mehdi Charles (DecisionBrain), mehdi.charles@decisionbrain.com ;
- Mathieu Lerouge (Laboratoire MICS, CentraleSupélec), mathieu.lerouge@centralesupelec.fr.

1 Introduction

DecisionBrain est une startup qui développe des solutions d'optimisation et d'aide à la décision pour la logistique, l'industrie, la planification de main d'œuvre et la chaîne d'approvisionnement (*supply chain*) sur tous les continents. Nous vous proposons de vous confronter à une version simplifiée d'un projet que DecisionBrain a réalisé pour un client : un outil de planification d'une équipe de techniciens mobiles, initialement prévue pour créer les routes quotidiennes de 500 techniciens devant réaliser quelques 10 000 tâches.

Supposez donc que vous rejoignez l'équipe *Optimisation*. L'entreprise *TuttoBene* fait appel à nous afin de lui développer un logiciel d'aide à la décision. L'activité de cette entreprise consiste à envoyer des techniciens chez des clients (particuliers ou entreprises) pour réaliser des tâches (e.g. entretien de chaudière, réparation électrique, chauffage, ...). *TuttoBene* emploie un planificateur, chargé de définir les routes quotidiennes des techniciens. Ce planificateur a besoin d'un logiciel d'aide à la décision pour l'assister dans la planification de l'équipe mobile de techniciens. Votre travail consiste à développer l'algorithme d'optimisation sur lequel s'appuie le logiciel - d'autres collègues s'occupent de la partie interface du logiciel.

2 Description du problème étudié

Votre contact chez *TuttoBene* précise l'ensemble des considérations de son problème de planification lors d'une première réunion de travail. Le problème est défini sur un horizon d'une journée. Chaque tâche doit être réalisée en un lieu donné, pendant une fenêtre de temps donnée, et requiert un certain niveau de compétence. Chaque technicien est disponible pendant une fenêtre de temps qui correspond à sa journée de travail. Il peut avoir des périodes d'indisponibilité pendant sa journée de travail, qui sont imposées par son contrat (e.g. une pause déjeuner de 1h entre 12h et 14h) ou exceptionnelles (e.g. un rdv médical). Il a par ailleurs un niveau de compétence. Les techniciens se déplacent du lieu d'une tâche à un autre dans leur journée pour travailler. Ils quittent leur domicile et y reviennent en fin de journée. Une tâche ne peut être affectée à un technicien que s'il possède la compétence requise, à un niveau au moins équivalent. L'objectif du problème est d'affecter le maximum de tâches aux routes des techniciens à coût opérationnel minimum.

Notes : On considère que la distance entre deux points donnés en coordonnées latitude-longitude peut être approchée par leur distance sphérique ([page wikipedia de la trigonométrie sphérique](#)). De plus, on considère que les techniciens se déplacent en voiture à une vitesse moyenne de 50km/h.

3 Données

Les fichiers d'instances sont des fichiers Excel et forment des jeux de données disponibles sur la [page Edunao du projet](#). Chaque fichier d'instance se compose de 4 feuilles, telles que représentées sur les Figures 1 à 4 pour l'instance *InstanceBordeauxV2*. La feuille *Employees* liste les employés disponibles dans l'équipe mobile. Pour chacun est indiqué son prénom, les coordonnées GPS de son lieu de résidence, son domaine et son niveau de compétence et les horaires de début et de fin de sa journée de travail. La feuille *Employees Unavailabilities* donne, pour chaque employé concerné, les horaires de ses périodes d'indisponibilités et les coordonnées GPS du lieu où il se trouvera pendant ce temps. La feuille *Tasks* précise, pour chaque tâche, son numéro, les coordonnées GPS du lieu où la réaliser, sa durée, le domaine de compétence et le niveau requis et les horaires d'ouverture pendant

lesquels elle peut être réalisée. Enfin, la feuille *Tasks Unavailabilities* indique, pour chaque tâche concernée, chaque créneau pendant lequel elle ne peut pas être effectuée, à l'intérieur des horaires données dans la feuille *Tasks*. Vous devrez implémenter une fonction pour lire les instances et les donner à votre algorithme. Nous vous invitons dans votre code à créer et utiliser des classes (e.g. classe Ressource, classe Tache, ...) et des dictionnaires pour stocker les données et les résultats.

Note : attention 12:00pm correspond à 12h et non à minuit !

EmployeeName	Latitude	Longitude	Skill	Level	WorkingStartTime	WorkingEndTime
Valentin	45.15121765523164	-0.822092647754919	Oenology	2	8:00am	6:00pm
Ambre	44.93630928188769	-0.8309410298001655	Oenology	1	8:00am	6:00pm

Figure 1: Exemple de feuille *Employees* - InstanceBordeauxV2

EmployeeName	Latitude	Longitude	Start	End
Ambre	44.93630928188769	-0.8309410298001655	3:00pm	6:00pm

Figure 2: Exemple de feuille *Employees Unavailabilities* - InstanceBordeauxV2

TaskId	Latitude	Longitude	TaskDuration	Skill	Level	OpeningTime	ClosingTime
T1	44.556549383420084	-0.31939224223757195	60	Oenology	1	8:00am	6:00pm
T2	44.967500952177986	-0.6086852638150881	60	Oenology	1	8:00am	6:00pm
T3	45.14421541464031	-0.7342570469020379	60	Oenology	2	8:00am	6:00pm
T4	45.264808304867096	-0.7717887212411139	60	Oenology	2	8:00am	6:00pm
T5	45.044422793402624	-0.6687606009488057	60	Oenology	1	8:00am	6:00pm
T6	45.19957452440505	-0.7462077931750715	60	Oenology	1	8:00am	6:00pm
T7	45.397697776585	-0.9668192708194538	60	Oenology	2	8:00am	6:00pm
T8	45.023479086796385	-0.8072126299796225	60	Oenology	2	8:00am	6:00pm
T9	45.29291368453335	-0.9365361007032235	60	Oenology	1	8:00am	6:00pm
T10	45.08146166752168	-0.8062453230620741	60	Oenology	2	8:00am	6:00pm

Figure 3: Exemple de feuille *Tasks* - InstanceBordeauxV2

TaskId	Start	End
T2	8:00am	9:00am

Figure 4: Exemple de feuille *Tasks Unavailabilities* - InstanceBordeauxV2

4 Travail à réaliser

Pour chaque rendu des phases décrites dans les sections ci-après, nous attendons les documents suivants.

- Vous devez fournir les solutions que vous avez obtenues en résolvant les instances demandées. Chaque solution doit être donnée sous la forme d'un fichier *.txt*, intitulé *Solution<Place>V<X>ByM<Y>.txt*, où *Place* désigne le lieu (e.g. *Bordeaux*), *X* le numéro du jeu de données (e.g. 1, 2, ou 3) et *Y* le numéro de la méthode de résolution (e.g. 1, 2, ou 3). Le contenu doit être tel que représenté en Figure 5.

La première partie de ce fichier décrit les affectations des tâches. Chaque ligne renseigne les champs :

- *taskId*, correspondant à l'identifiant de la tâche ;
- *performed*, prenant pour valeur 1 si la tâche est réalisée, 0 sinon ;
- *employeeName*, correspondant le prénom de l'employé ;
- *startTime*, indiquant l'instant (en minutes, compris entre 0 et 1440) de début de la tâche.

La seconde partie, séparée de la précédente par une ligne vide, décrit l'instant de la pause de chaque employé.

Chaque ligne renseigne les champs :

- *employeeName*, correspondant au nom de l'employé ;
- *lunchBreakStartTime*, précisant l'instant (en minutes, compris entre 0 et 1440) de début de sa pause.

Dans le fichier, les informations sont séparées par des point-virgule ";". Il est important de respecter ce format pour que nous puissions vérifier vos solutions. La solution décrite dans le fichier de la Figure 5 est illustrée sur la Figure 6, dans laquelle les pauses des employés y sont indiquées par des étoiles vertes.

```
taskId;performed;employeeName;startTime;
T1;0;;
T2;1;Ambre;921;
T3;1;Valentin;648;
T4;1;Ambre;812;
T5;1;Valentin;787;
T6;1;Valentin;576;
T7;1;Ambre;664;
T8;1;Ambre;491;
T9;1;Valentin;496;
T10;1;Ambre;563;

employeeName;lunchBreakStartTime;
Valentin;727;
Ambre;724;
```

Figure 5: Exemple de fichier de solution - *SolutionBordeauxV2ByM2*

- Vous devez rédiger un rapport - que vous complétez au fur et à mesure des phases du projet - dans lequel vous décrivez vos modèles, vos algorithmes, vos résultats (tableaux, graphiques, etc.), vos analyses etc. Pensez à indiquer la machine (processeur, mémoire vive) utilisée pour résoudre les instances. Un fichier Jupyter Notebook ne constitue pas un rapport.
- Enfin, vous devez fournir votre code sous la forme :
 - soit d'un fichier Python *.py*, auquel cas vous décrivez succinctement les éléments importants de votre code dans votre rapport ;
 - soit d'un Jupyter Notebook incluant votre code et quelques éléments de description.

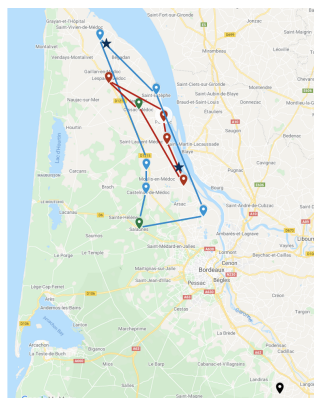


Figure 6: Solution associée au fichier *.txt* en Figure 5 - *SolutionBordeauxV2ByM2*

4.1 Phase 1 - Premier modèle exact simplifié

Pour le premier livrable, vous considérez une version simplifiée du problème prenant compte des caractéristiques les plus importantes - précisées ci-après. Les données de cette partie vous permettent de supposer que l'équipe de techniciens est suffisante pour réaliser toutes les tâches.

1. Proposer un premier Programme Linéaire en Nombre Entier (PLNE) modélisant une version simplifiée du problème étudié. La modélisation doit tenir compte des considérations suivantes :
 - l'objectif est de minimiser les coûts de transports ;
 - chaque employé part en début de journée et rentre en fin de journée en un lieu qui lui est propre ;
 - certains employés peuvent avoir des périodes d'indisponibilité dans la journée, associées à des lieux donnés (les employés doivent se trouver au lieu donné au début de l'indisponibilité et le quitter à la fin de celle-ci) ;
 - les pauses midi des employés ne sont pas considérées ;
 - les lieux des tâches sont ouverts sur un seul créneau dans la journée ;
 - les tâches ne peuvent être affectées qu'aux employés ayant un niveau de compétence suffisant.

Définir un tel PLNE suppose choisir les variables de décisions, en donner l'interprétation, de formuler les contraintes de façon mathématique etc. Plusieurs modélisations sont possibles.

2. Implémenter ce premier modèle PLNE en un programme Python + Gurobi. Exécuter le programme sur les instances du jeu *InstancesV1*. Présenter et représenter les solutions obtenues, les résultats (valeur d'objectif, temps de résolution, etc.). Observer la difficulté à résoudre les instances les plus grosses, en particulier *InstanceFinlandV1*.

A terminer pour le premier rendu intermédiaire (vendredi 03 mars 2023).

4.2 Phase 2 - Deuxième modèle exact étendu

Pour le second livrable, vous décidez de complexifier le modèle pour tenir compte de toutes les demandes du client. Dans cette partie, l'équipe de techniciens n'est plus nécessairement suffisante pour réaliser toutes les tâches sur une journée.

3. Proposer un deuxième PLNE modélisant une version plus complète du problème étudié. Les considérations supplémentaires (par rapport au premier modèle PLNE) sont les suivantes :
 - l'objectif est de maximiser la durée totale des tâches réalisées à coût opérationnel minimum ;
 - chaque employé possède une pause midi d'une durée d'1h à placer entre 12h et 14h, dans un lieu quelconque ;
 - les lieux de certaines tâches peuvent être ouverts sur plusieurs créneaux disjoints dans la journée.

Cette deuxième modélisation PLNE peut demander à faire d'autres choix de variables de décision, de contraintes etc.

4. Implémenter ce deuxième modèle PLNE en Python + Gurobi. Exécuter le programme sur les instances du jeu *InstancesV2*. Observer la difficulté à résoudre le problème de façon exacte quand la taille des instances augmente - en particulier *InstanceSpainV2*, *InstanceAustraliaV2*. Comparer les temps de calculs obtenus entre les deux versions du modèle sur des instances du jeu *InstancesV1*.
5. Analyser la qualité des solutions obtenues au moyen d'indicateurs de performance. Proposer des modifications du modèle afin d'améliorer ces performances.

A terminer pour le deuxième rendu intermédiaire (mercredi 15 mars 2023).

4.3 Phase 3 - Résolution approchée

Constatant qu'il est difficile de résoudre les instances réelles de façon exacte, vous décidez de développer une méthode de résolution approchée.

6. Construire une métaheuristique capable de calculer une bonne solution dans des temps raisonnables (quelques minutes) pour le jeu de données réel *InstancesV3*. Afin d'évaluer la qualité des solutions obtenues avec la métaheuristique, il peut être pertinent de résoudre aussi les petites instances et de comparer les résultats obtenus avec les solutions optimales.
7. Pour aller plus loin, vous pouvez aussi explorer différentes idées.
 - Vous pouvez étudier l'impact du choix de l'objectif, en utilisant différentes formulations multi-objectifs, en faisant varier les poids des différents sous-objectifs etc. et en analysant la qualité des solutions obtenues. Vous pouvez par exemple introduire la notion d'équité dans les charges de travail.
 - Comment pouvez-vous encore améliorer les performances de votre métaheuristique (qualité des solutions, temps de calculs...) ?
 - Vous pouvez chercher une façon de mesurer l'écart à l'optimalité des solutions de votre métaheuristique. Vous pouvez pour cela rechercher une borne inférieure (en minimisation) ou supérieure (en maximisation) de la valeur optimale.
 - Vous pouvez considérer des contraintes additionnelles, par exemple des contraintes limitant la distance totale quotidienne par employé, des contraintes modélisant la préférence de certains employés pour certaines tâches ou la possibilité de demander des heures supplémentaires aux employés pour réaliser plus de tâches, dans des limites définies. Vous pouvez créer de nouvelles instances pour tester ces fonctionnalités, que vous incluez dans vos résultats.
 - Vous pouvez également plancher sur cette question que vous vous posez et que vous aimeriez étudier !

A terminer pour le rendu final (vendredi 31 mars 2023).

5 Conseils

Nous vous invitons à utiliser Git afin de travailler plus facilement ensemble.

Nous restons disponibles pour répondre à vos questions, sur le canal Teams de votre équipe également en dehors des cours. N'hésitez pas à nous solliciter !

6 Critères d'évaluation

L'évaluation porte d'une part sur des critères relatifs à la qualité de la production tels que

- la pertinence des modèles ;
- la qualité du code développé ;
- les performances obtenues ;
- la critique des modèles / algorithmes développés et des résultats obtenus ;
- la qualité des rendus.

Elle porte d'autre part sur des critères relatifs au processus de travail du groupe

- la prise d'initiatives ;
- le travail d'équipe.

7 Bibliographie

La planification d'une équipe mobile est un problème d'optimisation générique que l'on peut rencontrer dans divers contextes professionnels. Dans la littérature, ce problème générique est nommé *Workforce Scheduling and Routing Problem (WSRP)*. Il s'agit d'une généralisation du problème de routage de véhicule avec fenêtres de temps, *Vehicle Routing Problem with Time Windows (VRPTW)*, lui-même une extension du problème de voyageur de commerce, *Traveling Salesman Problem (TSP)*.

Ce domaine a été largement étudié depuis les premiers travaux de Begur et al. (1997) à la fin des années 90. Parmi les nombreux articles publiés, ceux de Castillo-Salazar et al. (2016), de Fikar and Hirsch (2017) et de Trautsamwieser and Hirsch (2011) sont particulièrement intéressants dans le contexte de ce projet.

Ces articles sont disponibles sur la [page Edunao du projet](#).

References

Begur, S.V., Miller, D.M., Weaver, J.R., 1997. An integrated spatial dss for scheduling and routing home-health-care nurses. *Interfaces* 27, 35–48.

Castillo-Salazar, J., Landa-Silva, D., Qu, R., 2016. Workforce scheduling and routing problems: literature survey and computational study. *Annals of Operations Research* 239.

Fikar, C., Hirsch, P., 2017. Home health care routing and scheduling: A review. *Computers & Operations Research* 77, 86 – 95.

Trautsamwieser, A., Hirsch, P., 2011. Optimization of daily scheduling for home health care services. *JAOR* 3, 124–136.