



CentraleSupélec

PROJET : SYSTÈME DE DÉCISION ET PRÉFÉRENCES

CENTRALESUPÉLEC

Matthieu Annequin - Valentin Odde - Arthur
Guédon

Projet modélisation planning

2023

Encadré par :

Vincent MOUSSEAU

Table des matières

1	Introduction et présentation du problème	1
2	Modélisation	1
2.1	Analyse du problème	1
2.2	Modèle choisi	1
2.3	Notations	2
2.3.1	Variables de décision	2
2.3.2	Contraintes	2
2.3.3	Fonctions objectifs	3
3	Génération d’instances	3
4	Résultats obtenus	4
4.1	Approche mono-objectif	4
4.2	Approche multi-objectifs	4
5	Conclusion	6
	Conclusion	6

1 Introduction et présentation du problème

Nous nous intéressons à la société *CompuOpti* et nous réalisons pour eux un modèle de création de planning pour leurs employés afin de les staffer sur des projets de leurs clients. Chaque projet demande un certain nombre de jours/personne pour différentes compétences. Il s'agit donc d'affecter de manière optimisée chaque employé, qui possède un ensemble de compétences, sur différents projets.

2 Modélisation

2.1 Analyse du problème

Il s'agit d'un problème d'affectation de projets et d'optimisation de planning de personnel. En entrée, nous disposons de données concernant :

- **Un horizon de temps** sur lequel nous devons réaliser le planning. Nous ne considérons ici que les jours ouvrés.
- **Les projets** : pour chaque projet, il y a un ensemble de jours de travail à effectuer pour chaque compétence nécessaire à la réalisation du projet. Un projet produit un gain financier lors de sa réalisation et dispose d'une date limite de rendu. Un projet a une pénalité financière qui s'applique pour chaque jour de retard de rendu du projet.
- **Les employés** : pour chaque employé, il y a un ensemble de compétences maîtrisées, et des jours de congés définis à l'avance.

Personnel		Projets			
nom	compétences		A	B	C
alice	A,B	I	4	3	2
bob	C	II	3	2	7
charlie	A	III	4	1	5
david	B	IV	3	3	3
eve	A,B,C	V	1	1	4

FIGURE 1 – Exemple de données d'entrée de projets et d'employés

Les objectifs de la réalisation de ce planning sont multiples :

- Maximiser le gain financier (produit lorsqu'un projet est terminé)
- Minimiser le nombre de projet différents sur lequel travaille un collaborateur.
- Minimiser le nombre de jours consécutifs sur lequel un projet est effectué.

2.2 Modèle choisi

Nous approchons ce problème avec une modélisation par la programmation linéaire. Il y aura 4 dimensions (temps, projet, employé, compétence). Nous avons d'abord fait une modélisation fonctionnelle (pour la petite instance) mais dont le temps de calcul était trop long pour les

instances plus grosses. Nous avons cherché à l'optimiser mais le temps de calcul restant trop long, nous nous sommes finalement rabattu sur la modélisation proposée par Vincent Mousseau.

2.3 Notations

Nous pouvons d'abord définir quelques notations utiles dans la suite de notre modélisation :

- $h \in \mathbb{N} \setminus \{0\}$ l'horizon de temps
- $n_{j,k}$ le nombre de jours/personnes à réaliser sur le projet j pour la compétence k
- g_j le gain financier obtenu lorsque le projet j est terminé
- c_j la pénalité financière pour le projet j par jour de retard

2.3.1 Variables de décision

Nous définissons les variables de décision suivantes :

- $X_{i,j,k,t} \in \{0, 1\}$ variable binaire qui vaut 1 si l'employé i travaille le jour t sur le projet j avec la compétence k , 0 sinon
- $Y_j \in \{0, 1\}$ variable binaire qui vaut 1 si le projet j est terminé, 0 sinon.
- L_j le nombre de jours de retard du projet j
- E_j la date à laquelle est terminée le projet j

2.3.2 Contraintes

Soit :

- h l'horizon de temps et $H = \llbracket 1; h \rrbracket$
- q le nombre de compétences et $Q = \llbracket 1; q \rrbracket$
- n le nombre d'employés et $S = \llbracket 1; n \rrbracket$
- m le nombre de projets et $J = \llbracket 1; m \rrbracket$,
- C_i l'ensemble des jours de congés de l'employé i
- Q_i^S l'ensemble des compétences de l'employé i
- Q_j^J l'ensemble des compétences nécessaires à la réalisation du projet j

Nous avons les contraintes suivantes à respecter :

- $\forall i \in S, \forall t \in H, \sum_{j=1}^m \sum_{k=1}^q X_{i,j,k,t} \leq 1$: chaque jour, un employé peut travailler au plus sur un projet avec une seule compétence.
- $\forall i \in S, \forall t \in C_i, \sum_{j=1}^m \sum_{k=1}^q X_{i,j,k,t} = 0$: les employés ne travaillent pas lors de leurs jours de congés.
- $\forall i \in S, \forall j \in J, \forall t \in H, \forall k \notin Q_i^S \vee k \notin Q_j^J, X_{i,j,k,t} = 0$: un employé ne travaille pas sur un projet avec une compétence qu'il ne possède pas ou qui n'est pas requise pour terminer le projet.
- $\forall j \in J, \forall k \in Q_j^J, Y_j \times n_{j,k} \leq \sum_{i=1}^n \sum_{t=1}^h X_{i,j,k,t}$: Pour chaque projet et pour chaque compétence d'un projet, il n'est pas possible de travailler sur cette compétence lorsque le projet est terminé.
- $\forall j \in J, \forall k \in Q_j^J, \sum_{i=1}^n \sum_{t=1}^h X_{i,j,k,t} \leq n_{j,k}$: Pour chaque projet et pour chaque compétence d'un projet, il n'est pas possible d'avoir travaillé plus que nécessaire pour terminer le projet

- $\forall i \in S, \forall j \in J, \forall k \in Q, \forall t \in H, X_{i,j,k,t} \times t \leq E_j$ Un employé ne peut pas travailler sur un projet plus longtemps que la durée prévue par la date de fin de réalisation du projet.

2.3.3 Fonctions objectifs

On cherche avant tout à maximiser le gain financier de l'entreprise. Cela revient donc à maximiser cette fonction :

$$\sum_{j \in J} (Y_j \times g_j - L_j \times c_j)$$

Le premier terme dans la somme correspond au gain obtenu par les projets terminés et le second aux pénalités pour les projets ayant été terminés avec des jours de retard.

La deuxième fonction objectif est le nombre de projet maximal par employé. On l'obtient à partir de la matrice X . Il faut ainsi compter pour chaque employé sur combien de projet différent il travaille et ensuite choisir le plus grand parmi ces derniers. On utilise alors une formulation linéaire de la fonction maximum qui est déjà implémenté dans Gurobi.

Enfin la dernière fonction objectif correspond à la durée du projet le plus long. On calcule alors la durée de chaque projet et on regarde la durée du plus long. Pour cela nous avons déjà dans nos variable la date de fin de chacun des projets. Nous introduisons donc une nouvelle variable qui va représenter pour chaque projet sa date de début. Nous pouvons ensuite obtenir sa durée en soustrayant.

Pour obtenir la date de début, nous utilisons un matrice projet x temps qui vaut 1 à partir du moment où un projet est commencé. On peut ensuite faire un calcul à partir de cette matrice pour obtenir la date de début.

3 Génération d'instances

Afin de disposer d'un plus grand nombre de données pour tester notre modélisation ainsi que l'algorithme de résolution, nous avons créé un programme permettant de générer de nouvelles instances. Ce programme prend en entrée quatre paramètres :

- Un horizon de temps
- Un nombre d'employé
- Un nombre de projets
- Un nombre de compétences

Avec ces paramètres d'entrée, nous générons de manière aléatoire les compétences maîtrisées par chaque employé ainsi que leurs jours de congés. De même nous assignons, pour chaque projet, le nombre de jours nécessaires par compétences pour sa réalisation, un gain, une pénalité de retard et une date de rendu limite.

La grande part d'aléatoire dans la génération de ces instances peut donner lieu à des projets irréalisables (trop de compétences requises sur un projet par rapport au nombre d'employés par exemple). Nous avons donc encadré la génération de ces paramètres avec des bornes permettant de respecter l'horizon de temps et les compétences disponibles.

L'instance générée est renvoyée sous la forme d'un dictionnaire au même format que les données de test déjà fournies.

4 Résultats obtenus

4.1 Approche mono-objectif

Pour l'approche mono-objectif, en utilisant cette modélisation on obtient le résultat suivant pour la petite instance.

	0	1	2	3	4
Olivia	(Job3, C)	(Job1, A)	(Job4, B)	(Job5, C)	(Job5, C)
Liam	None	(Job4, B)	(Job1, B)	(Job3, A)	None
Emma	(Job4, C)	None	(Job1, C)	(Job3, C)	None

FIGURE 2 – Résultat sur la petite instance

Il est alors très facile de vérifier que toutes les contraintes sont respectées. Pour ce résultat, on obtient un gain de 65. Pour les autres instances voici on fait également marcher notre modélisation. On obtient les gains suivants :

- Petite instance : 65
- Moyenne instance : 413
- Grande instance : 817

On remarque que plus l'instance est grosse plus le temps de calcul est long. Ce temps de calcul est dépendant de la machine utilisé pour le calculer. Etant donné que le problème est NP-difficile, il est normal de voir ce temps augmenter. On peut s'amuser avec notre générateur d'instance, à mesurer la taille de l'instance à partir de laquelle le problème devient trop "gros" pour être résolu avec nos machines et donc savoir jusqu'à quel (et donc savoir jusqu'à quelle taille la société *CompuOpti* peut planifier son personnel sur ses différents projets). Par exemple, si on génère une instance avec un horizon de 2 mois (60 jours), 20 collaborateurs, 30 projets et 15 compétences, le temps de calcul est supérieur à 1 heure pour notre machine. Cette réflexion autour du temps de calcul devra être prise en compte dans l'utilisation réelle de notre outil (nécessité de faire de la planification de sous-équipes pour les équipes trop grandes, planification sur des échelles de temps courtes, prévoir des ressources de calculs puissantes).

4.2 Approche multi-objectifs

Nous nous sommes ensuite intéressé à l'approche d'optimisation multi-objectif, où nous avons désormais 3 objectifs à accomplir :

- Maximiser le bénéfice financier de l'entreprise

- Minimiser le nombre de projets par collaborateurs. On souhaite que les collaborateurs changent le moins possible de projet.
- Minimiser la durée du plus long projet. En effet, on souhaite que les projets soient réalisés dans un minimum de nombre de jours consécutifs.

Pour obtenir la surface des solutions non-dominées du problème, nous avons appliqué l'algorithme d'optimisation mono-objectif en rajoutant des contraintes permettant de fixer le nombre maximum de projets par collaborateurs et la durée de réalisation du plus long projet. Nous faisons ensuite varier ces deux paramètres et appliquons l'algorithme de résolution à chaque itération. On remarque que ces deux valeurs sont des entiers bornés :

- Un collaborateur travaille sur au moins 1 projet et au plus sur tous les projets
- Le plus long projet est réalisé au bout d'au moins 1 jour et au maximum l'horizon de temps.

Sur la petite instance on obtient la surface de Pareto suivante :

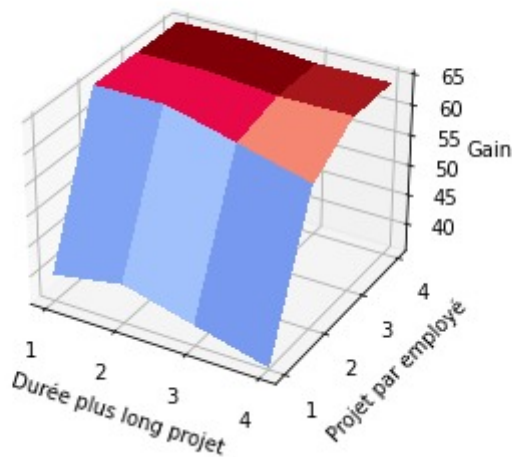


FIGURE 3 – Surface de Pareto de la petite instance

Nous obtenons ainsi en un temps très raisonnable une représentation explicite des courbes de Pareto de nos modèles. Cela est rendu possible par le fait que nous avons des fonctions objectives entières et bornées et donc nous pouvons pour chacune des valeurs possibles de ces entiers calculer explicitement le gain.

Avec cette surface nous disposons de toute l'information nécessaire pour pouvoir obtenir le planning idéal selon l'arbitrage traduit entre les fonctions objectifs. On peut ainsi par exemple savoir combien de gain il faut sacrifier pour obtenir que les collaborateurs ne travaillent pas sur plus de 3 projets par exemple.

On peut alors quantifier de manière précise auprès des décideurs ces informations, et obtenir un arbitrage entre les fonctions objectifs nous permettant de déterminer la solution la plus adaptée à la vision du dirigeant.

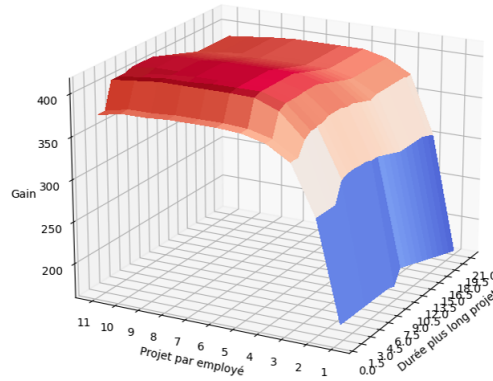


FIGURE 4 – Surface de Pareto de l'instance intermédiaire

5 Conclusion

Nous avons développé une solution permettant de générer des plannings optimisés pour affecter le personnel de la société *CompuOpti*, l'objectif de cette optimisation est avant tout de maximiser le gain de l'entreprise, mais nous prenons également en compte des objectifs de minimisation de la durée des projets et du nombre de projets différents sur lesquels travaillent les collaborateurs.

Il est alors nécessaire d'effectuer un arbitrage entre ces 3 critères puisqu'ils ne peuvent pas tous être optimisés simultanément. On pourrait alors imaginer un outil qui permettrait au dirigeant de piloter ces critères en fonction de la stratégie de l'entreprise, par exemple prioriser le bénéfice de l'entreprise ou plutôt la réalisation rapide des projets.

On note néanmoins une limite à notre outil qui ne peut pas s'adapter à des horizons de temps plus long ou à un nombre de projets trop important, et donc qui n'est pas adapté à des entreprises trop grandes.