

Systemes de D cision et Pr f rences : Projet

Pierre-Alexandre Klein, Juliette L orat, Daniel L vy

10 F vrier 2023

Contents

1	Introduction	3
2	Instances	3
3	Modélisation	4
3.1	Paramètres et Variables	4
3.2	Fonctions objectifs	6
3.3	Contraintes	6
4	Implémentation sur Gurobi	8
4.1	Modélisation	8
4.2	Epsilon-Constraint et Surface de Pareto	9
5	Résultats	11
5.1	Analyse des performances	11
5.2	Difficultés rencontrées	14
6	Annexe	15

List of Figures

1	Toy instance	4
2	Surface de Pareto	10
3	Points non-dominés	10
4	Outputs	12
5	Outputs	12
6	Planning pour la toy instance	12
7	Résultats	13
8	Planning pour la medium instance	15

1 Introduction

CompuOpti est une entreprise qui développe et implémente des solutions d'optimisation et d'aide à la décision pour ses clients. Le problème étudié est la planification du personnel et l'affectation des projets. Le personnel de CompuOpti a des qualifications parmi un ensemble donné, ainsi que des jours de congé prédéfinis. Chaque projet nécessite des qualifications spécifiques et produit un gain si réalisé. Le but est de maximiser le bénéfice financier de l'entreprise tout en prenant en compte d'autres critères tels que le nombre de projets sur lesquels un collaborateur est affecté et la durée des projets consécutifs.

2 Instances

Nous avons à disposition 3 jeux d'instances de tailles différentes qui sont au format json.

Chaque instance représente un cas avec des informations sur le contexte à savoir:

- l'**horizon**, le nombre de jours sur lesquels les projets pourront être réalisés
- les **qualifications**, l'ensemble des qualifications du jeu d'instances
- les **employés**, définis par :
 - Nom
 - Qualifications de l'employé
 - Jour de congés
- les **projets** à réaliser, définis par:
 - Nom
 - Gain obtenu si le projet est réalisé
 - Pénalité journalière
 - Due Date
 - Temps de qualifications nécessaire à la réalisation du projet

En plus des 3 instances qui nous été données (small, medium, large), nous avons créé l'instance test. Cette instance nous a permis de s'assurer de la bonne paramétrisation de nos variables et contraintes notamment en nous aidant à manipuler la matrice 4D du Planning.

L'instance test reprend la toy instance avec:

Horizon=5, Qualifications=3, Staff=2, Jobs=4

Voici un extrait de la toy instance :

```

{
  "horizon": 5,
  "qualifications": [
    "A",
    "B",
    "C"
  ],
  "staff": [
    {
      "name": "Liam",
      "qualifications": [
        "A",
        "B"
      ],
      "vacations": [
        1
      ]
    }, ...
  ],
  "jobs": [
    {
      "name": "Job1",
      "gain": 20,
      "due_date": 3,
      "daily_penalty": 3,
      "working_days_per_qualification": {
        "A": 1,
        "B": 1,
        "C": 1
      }
    }, ...
  ]
}

```

Figure 1: Toy instance

3 Modélisation

3.1 Paramètres et Variables

Afin de modéliser le problème d'optimisation, il est important de définir les paramètres et les variables qui rentrent en jeu.

$H = \{1, \dots, h\}$, avec $h \in N \setminus \{0\}$ horizon de temps

$Q = \{1, \dots, q\}$, l'ensemble des qualifications

$S = \{1, \dots, s\}$, est l'ensemble du personnel

$P = \{1, \dots, p\}$ est l'ensemble des projets

$s \in S$: staff

$h \in H$: horizon

$p \in P$: project

$q \in Q$: qualification

D_p : Due date of the project p

$$1 \leq D_p \leq h \quad (1)$$

B_p : Begin date of the project p

$$1 \leq B_p \leq h \quad (2)$$

E_p : End date of the project p

$$1 \leq E_p \leq h \quad (3)$$

L_p : Delay (number of days late for the project p)

$$0 \leq L_p \leq h \quad (4)$$

V_s : Vacation (days off for the member of staff s)

$$1 \leq V_s \leq h \quad (5)$$

$N_{pq} \in N$: Number of days/people per qualification q for project p

R_p : Realization of the project p

$$R_p = \begin{cases} 1 & \text{si projet est réalisé,} \\ 0 & \text{sinon} \end{cases} \quad (6)$$

$P_p \in N$: Penalty (per day of delay for the project p)

$G_p \in N$: Gain for the project p

$Le_p \in N$: Length of a project p

$MaxD \in N$: Max duration for a project p

$JobStaffeur_{s,p}$: Matrice d'affectation $\forall s \in S, \forall p \in P$

$SumStaffJobs_p$: Nombre de projet par employé

X_{shpq} : Planning (indicates whether a member of staff s is employed to perform a qualification q for project p on day h)

$$X_{shpq} = \begin{cases} 1 & \text{si } s \text{ réalise } q \text{ de } p \text{ au jour } h, \\ 0 & \text{sinon} \end{cases} \quad (7)$$

3.2 Fonctions objectifs

Pour élaborer les emplois du temps des membres du personnel, c'est-à-dire affecter chaque jour de travail d'un membre du personnel à une qualification d'un projet, nous devons définir des critères selon lesquels nous allons optimiser les plannings. L'objectif du projet est de trouver le meilleur compromis entre ces critères.

Il y a trois objectifs que CompuOpti souhaite optimiser:

Tout d'abord, il s'agit de maximiser le résultat financier de l'entreprise et donc de constituer un planning qui conduit à maximiser le bénéfice, en prenant en compte les éventuelles pénalités qui peuvent être causées par le retard pris par chaque projet. La fonction objectif Z_1 peut être modélisée de la façon suivante:

Z_1 :

$$\max\left(\sum_p (R_p \cdot G_p - P_p \cdot L_p)\right) \quad (8)$$

Ensuite, il s'agit de minimiser le nombre de projets sur lesquels un collaborateur est affecté pour réduire les changements fréquents de projet. La fonction objectif Z_2 s'écrit alors:

Z_2 :

$$\min(\text{nombre projets par employés}) \quad (9)$$

$$\min\left(\sum_s \sum_p X_{shpq}\right) \quad (10)$$

Enfin, on cherchera à exécuter les projets les plus longs en un minimum de jours consécutifs pour minimiser la durée de réalisation des projets. La fonction objectif Z_3 se traduit par:

Z_3 :

$$\min(\text{durée du projet le plus long}) \quad (11)$$

$$\min(\max(E_p - B_p)) \quad (12)$$

3.3 Contraintes

Pour la constitution du planning, un certain nombre de contraintes sont à respecter :

Contrainte 1 : Unicité de l'affectation quotidienne du personnel

A tout instant, un membre du personnel ne peut être affecté qu'à un seul projet et qu'à une seule qualification intervenant dans ce projet.

$$\sum_{(p,q)} X_{s,p,q,h} \leq 1 \quad \forall s \in S, \forall h \in H \quad (13)$$

Contrainte 2 : Congés

Un membre du personnel ne peut pas être affecté à une qualification de projet un jour de congé.

$$\sum_{(p,q)} X_{s,p,q,h} = 0 \quad \forall s \in S, \forall h \in V_s \quad (14)$$

Contrainte 3 : Qualification du personnel

Un membre du personnel ne peut être affecté à une qualification d'un projet que s'il possède cette qualification.

$$X_{s,p,q,h} = 0 \quad \forall s \in S, \forall p \in P, \forall q \in Q \mid q \notin Q_s \vee q \notin Q_p, \forall h \in H \quad (15)$$

Contrainte 4 : Couverture des qualifications du projet

Un projet n'est considéré réalisé que si tous les jours de travail dédiés à chacune des qualifications intervenant dans le projet ont été couverts par des membres du personnel.

$$R_p * N_{p,q} \leq \sum_{(s,h)} X_{s,p,q,h} = 0 \quad \forall p \in P, \forall q \in Q \quad (16)$$

Contrainte 5 : Unicité de réalisation d'un projet

Un projet ne peut être réalisé qu'une fois sur une période de temps donnée.

$$\sum_{(s,h)} X_{s,p,q,h} \leq N_{p,q} \quad \forall p \in P, \forall q \in Q \quad (17)$$

Contrainte 6 : Date de fin d'un projet

Le nombre de jours travaillés sur un projet ne peut excéder la date de fin de ce projet.

$$X_{s,p,q,h} * h \leq E_p \quad \forall s \in S, \forall p \in P, \forall q \in Q, \forall h \in H \quad (18)$$

Contrainte 7 : Jours de retard

La différence entre la date de fin (End date) et la date d'échéance (Due date) doit être inférieure ou égale à la limite de retard (Delay) permise.

$$E_p - D_p \leq L_p \quad \forall p \in P \quad (19)$$

Contrainte 8 : Un projet ne peut commencer avant le jour 1

$$B_p \geq 1 \quad \forall p \in P \quad (20)$$

Contrainte 9 : Liste d'affection d'un employé sur les projets

$$JobStaffeur_{s,p} == \max(val) \quad \forall s \in S, \forall p \in P, \forall q \in Q, \forall h \in H \quad (21)$$

ou val =

$$[X_{s,p,q,h}] \quad (22)$$

Contrainte 10 : Somme des projets réalisés par un employé est égal au total des projets réalisés par cet employé

$$SumStaffJobs_p == \sum JobStaffeur_p \quad \forall p \in P \quad (23)$$

Contrainte 11 : Date de début d'un projet

$$B_p \leq X_{s,p,q,h} \cdot (h + 1) + h \cdot (1 - X_{s,p,q,h}) \quad \forall s \in S, \forall p \in P, \forall q \in Q, \forall h \in H \quad (24)$$

Contrainte 12 : Durée d'un projet

$$E_p + 1 - B_p \leq Le_p \quad \forall p \in P \quad (25)$$

Contrainte 13 : Durée max d'un projet

$$Le_p \leq MaxD \quad \forall p \in P \quad (26)$$

4 Implémentation sur Gurobi

4.1 Modélisation

Nous avons utilisé le solveur Gurobi afin de résoudre notre problème d'optimisation sous contraintes. Les variables modélisées précédemment ont été ajoutées au modèle créé sur Gurobi, certaines étant spécifiques à un critère en particulier.

Variabes posées dans Gurobi :

- Gm : gain
- Pm : penalty
- Dm : due date
- Delay
- BeginProject
- EndDate
- Planning : correspond à la variable X
- Vacations : jours de congés
- QualificationsStaff : qualifications que possède chaque employé
- QualificationsJob : qualifications requises pour un projet
- Realization
- JobsStaffeur : projets réalisés pour chaque employé
- SumStaffJobs : total des projets réalisés
- LengthProject : durée d'un projet
- MaxDurationLength : durée du plus long projet

Nous avons ensuite implémenté les contraintes :

- Pour le critère n°1 les contraintes 1 à 8 sont concernées.
- Pour le critère n°2 on ajoute les contraintes 9 et 10
- Pour le critère n°3 on ajoute les contraintes 11, 12, 13

4.2 Epsilon-Constraint et Surface de Pareto

Dans le cadre de ce problème d'optimisation multi-objectif, nous appliquons la méthode e-constraint qui consiste à trouver un compromis entre plusieurs critères en utilisant une marge d'erreur e . Cette marge est utilisée pour tenir compte de la variabilité de la solution optimale et pour permettre de déterminer les solutions acceptables en fonction des critères d'optimisation. En particulier, la méthode epsilon-constraint permet de trouver les solutions non-dominées dans un espace multidimensionnel, notamment les points de la surface de Pareto, qui représentent les compromis optimaux entre les différents critères d'optimisation.

Nous choisissons de partir de la première fonction objectif Z1 qui cherche à maximiser l'objectif financier : $\text{Max} (\text{Sum}(\text{Rm} \cdot \text{Gm}) - (\text{Pm} \cdot \text{Lm}))$

Nous utilisons les deux autres fonctions objectives comme contraintes en utilisant la valeur d'epsilon comme limite. La valeur d'epsilon est modifiée à chaque itération et il y a 2 boucles imbriquées. Les plages de valeurs des objectifs Z2 et Z3 sont des nombres entiers donc les valeurs d'epsilon le sont aussi ce qui permet d'obtenir une solution optimale.

Surface de Pareto et points non dominés:

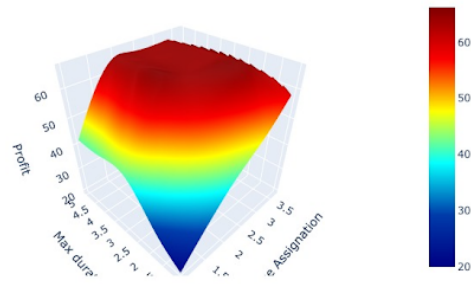


Figure 2: Surface de Pareto

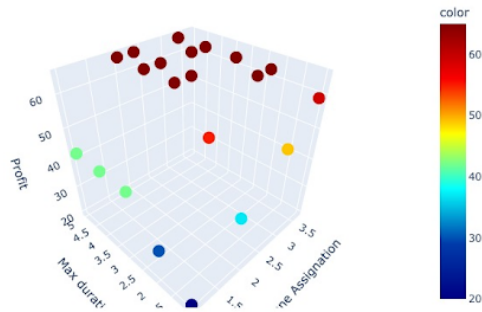


Figure 3: Points non-dominés

5 Résultats

5.1 Analyse des performances

Après l'exécution du modèle optimisé avec Gurobi, nous avons cherché à interpréter les résultats de l'optimisation.

Voici quelques éléments d'interprétation de la solution :

Statut de la solution : le statut de la solution peut être un indicateur important de la qualité de la solution obtenue. Les statuts les plus courants sont "Optimal", "Infeasible", "Unbounded" et "Inconnu".

Valeur optimale : la valeur optimale est le résultat de l'optimisation qui correspond à la fonction objective. C'est la valeur qui minimise ou maximise la fonction objective en fonction des contraintes.

Variables optimales : les variables optimales sont les valeurs des variables du modèle qui correspondent à la valeur optimale. Elles peuvent être utilisées pour comprendre les choix faits par l'optimiseur.

Contraintes : il est important de vérifier si les contraintes du modèle sont respectées par la solution optimale. Si une contrainte n'est pas respectée, il peut être nécessaire de la revoir ou de la reformuler.

Temps de calcul : le temps de calcul peut être utilisé pour évaluer la rapidité de la résolution du modèle et peut être utilisé pour déterminer si des améliorations peuvent être apportées à la mise en œuvre pour accélérer le temps de calcul.

Pour cela nous avons commencé par récolter et comparer les solutions optimales que nous obtenions pour évaluer la performance.

- Z_1 : Le résultat financier est accessible via model.objVal
- Z_2 : La minimisation du nombre de projets différents par employé se trouve dans la variable SumStaffJobs
- Z_3 : La minimisation du temps du projet le plus long se retrouve via la soustraction des variables EndProject et BeginProject

<p>E1: Maximize Profit</p> <p>Solution Best</p> <p>65.0 80</p> <p>Details: Gain = Delay * Daily</p> <p>Penalty</p> <p>Job1 : 20 - 0 * 3</p> <p>Job2 : Not Done</p> <p>Job3 : 15 - 0 * 3</p> <p>Job4 : 20 - 0 * 3</p> <p>Job5 : 10 - 0 * 3</p> <p>*****</p> <p>Total = 65.0</p>	<p>E2: Minimize Number of Projects per Employee</p> <table><tr><th>Staff</th><th>Job(s)</th><th>Name</th></tr><tr><td>0</td><td>2</td><td>Olivia</td></tr><tr><td>1</td><td>2</td><td>Liam</td></tr><tr><td>2</td><td>2</td><td>Emma</td></tr></table>	Staff	Job(s)	Name	0	2	Olivia	1	2	Liam	2	2	Emma	<p>E3: Minimize Realization Time</p> <table><tr><th>Job</th><th>Begin</th><th>End</th><th>Delta</th></tr><tr><td>1</td><td>0</td><td>3</td><td>3</td></tr><tr><td>2</td><td>0</td><td>0</td><td>0</td></tr><tr><td>3</td><td>3</td><td>4</td><td>2</td></tr><tr><td>4</td><td>0</td><td>3</td><td>3</td></tr><tr><td>5</td><td>4</td><td>5</td><td>2</td></tr></table>	Job	Begin	End	Delta	1	0	3	3	2	0	0	0	3	3	4	2	4	0	3	3	5	4	5	2
Staff	Job(s)	Name																																				
0	2	Olivia																																				
1	2	Liam																																				
2	2	Emma																																				
Job	Begin	End	Delta																																			
1	0	3	3																																			
2	0	0	0																																			
3	3	4	2																																			
4	0	3	3																																			
5	4	5	2																																			

Figure 4: Outputs

Nous avons fait en sorte d’afficher les variables mises à jour par Gurobi. L’objectif était de vérifier la satisfaction des contraintes que nous avons définies.

<p>- Couverture Qualifications -</p> <p>Job1 True</p> <p>. A 1</p> <p>. B 1</p> <p>. C 1</p> <p>Expected [1 1 1]</p> <p>---</p> <p>Job2 False</p> <p>. A 0</p> <p>. B 0</p> <p>. C 0</p> <p>Expected [1 2 0]</p>	<p>- Due Dates -</p> <table><tr><th>Project</th><th>Expected</th><th>End</th><th>Delay</th></tr><tr><td>1</td><td>3</td><td>3</td><td>0</td></tr><tr><td>2</td><td>3</td><td>0</td><td>0</td></tr><tr><td>3</td><td>4</td><td>4</td><td>0</td></tr><tr><td>4</td><td>3</td><td>3</td><td>0</td></tr><tr><td>5</td><td>5</td><td>5</td><td>0</td></tr></table>	Project	Expected	End	Delay	1	3	3	0	2	3	0	0	3	4	4	0	4	3	3	0	5	5	5	0	<p>- Unicité Affectation et Congés -</p> <p>Jour 1</p> <p>. Olivia 1</p> <p>. Liam 0 Vacation</p> <p>. Emma 1</p> <p>---</p> <p>Jour 2</p> <p>. Olivia 1</p> <p>. Liam 1</p> <p>. Emma 0 Vacation</p> <p>---</p>
Project	Expected	End	Delay																							
1	3	3	0																							
2	3	0	0																							
3	4	4	0																							
4	3	3	0																							
5	5	5	0																							

Figure 5: Outputs

Nous avons créé une fonction qui nous permet de modéliser le Planning correspondant à la solution optimale grâce à la librairie matplotlib. Cette représentation nous a permis de se rendre compte rapidement des changements apportés par les modifications d’implémentation. Le planning medium instance est disponible en

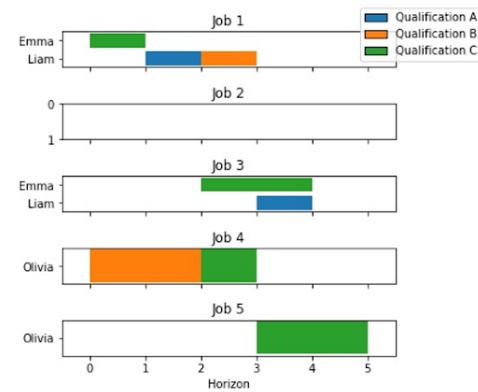


Figure 6: Planning pour la toy instance

annexe. Nous avons aussi pris soin de mesurer le temps d’exécution pour chaque instance. Avec plus de temps nous aurions aimé être en mesure d’apporter des modifications d’implémentation pour optimiser le temps d’exécution qui est très

Instance	Z1 : Profit	Z2 : Moyenne Assignment	Z3 : Durée Max	Runtime	Profit max
Toy	65	2.0	3	3s	80
Medium	413	4.4	12	1h17	510
Large Not Finished	817	7.0	11	>15h	1020

Figure 7: Résultats

long pour les grandes instances. Les principaux résultats sont résumés dans le tableau ci-dessus.

5.2 Difficultés rencontrées

Au cours de ce projet, plusieurs difficultés ont été rencontrées. Tout d’abord, la modélisation mathématique initiale s’est avérée complexe, mais des éléments de correction ont apporté une aide précieuse pour éclaircir les concepts. De plus, la transition de la modélisation mathématique au code Gurobi a été mise à rude épreuve en raison de la documentation insuffisante de Gurobi. Par conséquent, les séances de projets ont été cruciales pour comprendre les différentes étapes du processus. En outre, la manipulation de matrices 4D et la compréhension de la surface de Pareto ont également posé des défis. Enfin, le temps de calcul, en particulier pour les jeux d’instances medium et large ont été particulièrement long ce qui a freiné la progression du projet.

6 Annexe

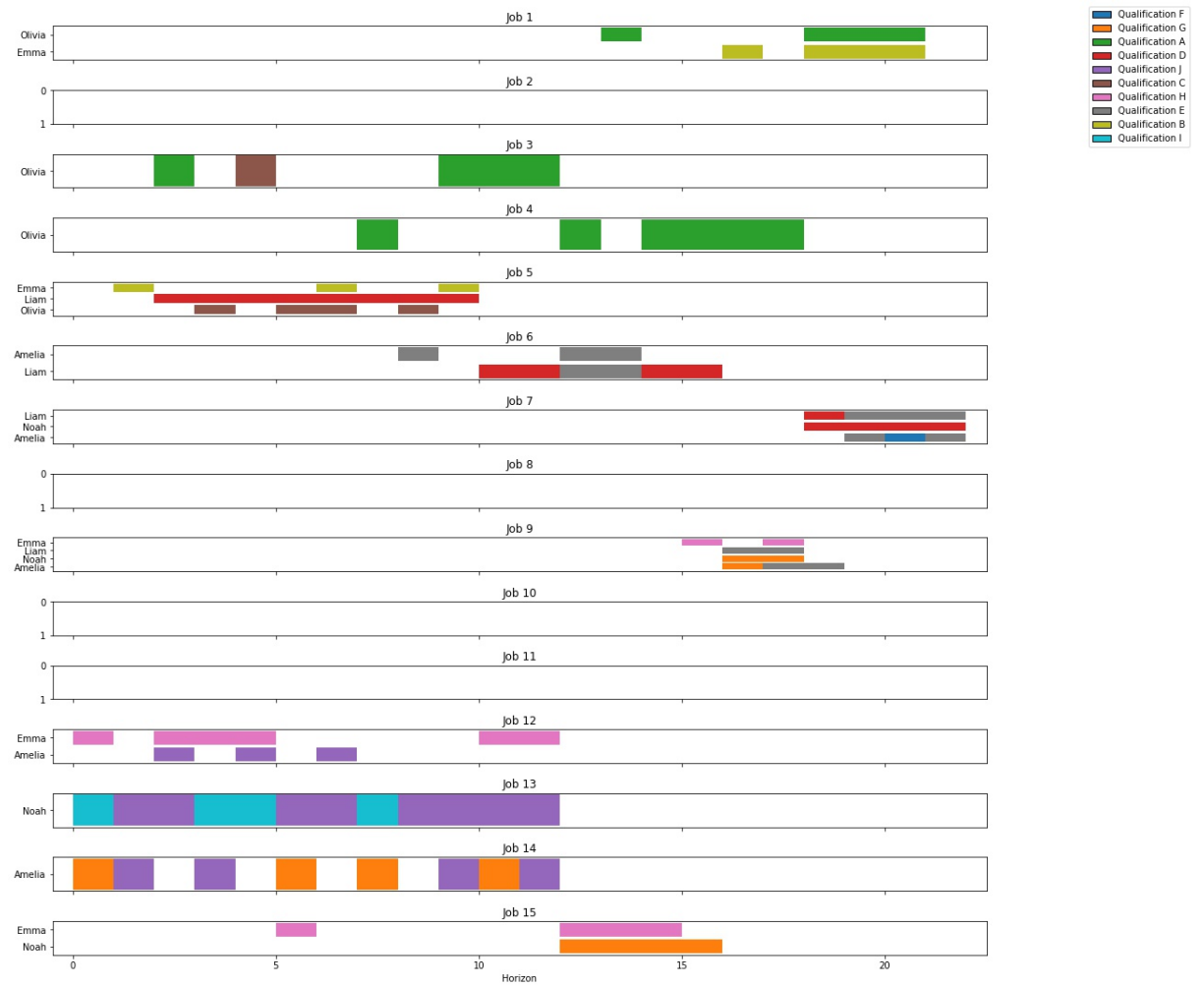


Figure 8: Planning pour la medium instance