Optimisation des emplois du temps des cours: Une approche basée sur les modèles logiques

Maxime Jauroyon
Philippe Masouf

Abstract

Ce rapport présente une approche basée sur les modèles logiques pour la gestion et l'optimisation des emplois du temps des cours. Nous développons et analysons trois modèles logiques successifs, abordant différents aspects du problème tout en respectant les contraintes et les objectifs spécifiques aux établissements d'enseignement. Les résultats obtenus montrent que notre approche est capable de générer des emplois du temps de haute qualité.

1. Introduction

La gestion et l'optimisation des emplois du temps des cours est un problème complexe et difficile à résoudre en raison du grand nombre de contraintes et de préférences à prendre en compte. Les établissements d'enseignement doivent attribuer les horaires de cours, les salles et les enseignants tout en tenant compte des exigences pédagogiques, des disponibilités des enseignants et des étudiants, et des ressources matérielles disponibles.

Les emplois du temps bien organisés offrent de nombreux avantages, tels que l'amélioration de la qualité de l'enseignement, la réduction des conflits d'horaire et la maximisation de l'utilisation des ressources. Cependant, la complexité du problème rend difficile l'obtention de solutions optimales en utilisant des méthodes manuelles ou heuristiques.

Dans ce rapport, nous présentons une approche basée sur les modèles logiques pour résoudre le problème d'optimisation des emplois du temps des cours. Nous utilisons un modèle de programmation linéaire pour formuler le problème et prenons en compte les contraintes et les objectifs spécifiques aux établissements d'enseignement.

Le rapport est organisé en quatre parties principales. Dans la première partie, nous présentons un modèle logique simple pour déterminer les horaires de cours. Dans la deuxième partie, nous étendons ce modèle pour prendre en compte les contraintes liées aux salles de classe. Dans la troisième partie, nous proposons un modèle qui vise à minimiser les "trous" dans les emplois du temps des étudiants et des enseignants. Enfin, dans la quatrième nous apportons de nouvelles améliorations pour avoir un emploi du temps plus agréable pour les élèves et les enseignants.

Nous conclurons le rapport en discutant des résultats obtenus et en proposant des perspectives pour les recherches futures.

2. Modèle simple

2.1. Données

Les instances fournies pour le modèle contiennent des informations essentielles sur les cours et les contraintes de l'emploi du temps. Les données comprennent :

- **Cours :** Chaque cours est identifié par un code unique et a une durée spécifiée en minutes. Les cours peuvent avoir plusieurs occurrences hebdomadaires.
- **Créneaux horaires :** La journée est divisée en créneaux de durée fixe (par exemple, 60 minutes). Le nombre de créneaux par jour et le nombre total de jours de la semaine sont également spécifiés.
- **Salles :** Les salles de cours sont identifiées par un code unique et ont une capacité déterminée par le nombre de places disponibles.
- **Contraintes :** Les contraintes peuvent inclure des exigences spécifiques, telles que l'indisponibilité des salles à certains moments de la journée, des conflits entre les cours qui ne doivent pas être programmés en même temps ou des contraintes sur le nombre d'occurrences hebdomadaires pour un cours.

Ces données sont utilisées pour déterminer la structure de base de l'emploi du temps et pour vérifier que les solutions générées par les modèles logiques respectent les contraintes et les objectifs définis.

2.2. Variables de décision

Dans notre modèle logique, nous utilisons deux principales variables de décision pour représenter l'affectation des cours aux jours et aux créneaux horaires.

2.3. Affectation des cours aux jours (planning_cours_jours)

La première variable de décision, nommée planning_cours_jours, est utilisée pour indiquer à quel jour de la semaine un cours spécifique doit être programmé. Elle est définie pour chaque cours et chaque occurrence hebdomadaire, c'est-à-dire pour chaque fois qu'un cours doit être programmé au cours d'une semaine. Cette variable de décision est exprimée sous la forme d'un tableau à deux dimensions, avec les cours en abscisse et les occurrences hebdomadaires en ordonnée.

Cette variable de décision permet de garantir que les cours sont répartis sur l'ensemble de la semaine en fonction des contraintes de disponibilité des professeurs et des salles, tout en respectant les exigences pédagogiques telles que les prérequis, les horaires fixes et les séances simultanées.

2.4. Affectation des cours aux créneaux horaires (planning_cours_creneaux)

La seconde variable de décision, nommée planning_cours_creneaux, est utilisée pour indiquer à quel créneau horaire un cours spécifique doit être programmé. Elle est également définie pour chaque cours et chaque occurrence hebdomadaire, et est exprimée sous la forme d'un tableau à deux dimensions similaire à celui de la variable de décision précédente.

Toutefois, cette variable de décision est représentée sous forme d'intervalle, avec une taille déterminée par la durée du cours et la durée du créneau. Cette représentation permet de prendre en compte les contraintes liées à la durée des cours et à leur répartition sur les différents créneaux horaires disponibles.

Ces deux variables de décision sont intégrées aux contraintes et aux objectifs du modèle logique, ce qui permet de générer des emplois du temps qui respectent les exigences spécifiques et minimisent les coûts ou les inconvénients pour les étudiants. En combinant ces variables de décision avec les données du problème et les contraintes définies, notre modèle logique offre une représentation flexible et robuste du problème d'emploi du temps.

2.5. Contraintes

Afin de résoudre le problème d'emploi du temps, nous avons défini un ensemble de contraintes à respecter. Dans cette section, nous présentons et expliquons ces contraintes.

Jours des séances: Les séances qui ont effectivement lieu doivent être planifiées sur des jours différents de 0, qui est utilisé comme un jour poubelle. En revanche, les séances qui n'ont pas lieu sont fixées au jour 0.

Enchaînement chronologique des cours: Les cours doivent être planifiés de manière à ce qu'ils s'enchaînent chronologiquement. Cela signifie que le cours 1 doit avoir lieu avant le cours 2. Lorsque les cours ont lieu le même jour, le cours 1 doit se terminer avant le début du cours 2.

Pause méridienne: Un cours ne peut pas commencer avant la pause méridienne et se terminer après celle-ci. Les séances qui débutent avant la pause méridienne doivent être égales ou moins longues que le nombre de créneaux restant avant la pause.

Cours d'un même parcours: Les cours appartenant à un même parcours ne doivent pas être planifiés en même temps. Si deux cours ont lieu le même jour, l'un d'entre eux doit se terminer avant le début de l'autre.

Cours communs et parcours d'une filière: Les cours communs d'une filière ne doivent pas être planifiés en même temps que les cours des parcours de la filière. Si deux cours ont lieu le même jour, l'un d'entre eux doit se terminer avant le début de l'autre.

Écart de jours entre les séances: Les séances d'un même cours doivent respecter un écart de jours déterminé par un intervalle [ecartMin, ecartMax]. Cet écart garantit un temps suffisant entre les séances pour les étudiants et les enseignants.

Séances fixées: Certaines séances sont fixées à des jours et des créneaux spécifiques. Ces séances doivent être planifiées conformément à ces contraintes.

Cours ayant les mêmes horaires: Certains cours doivent avoir les mêmes horaires. Dans ce cas, ils doivent être planifiés le même jour, avec les mêmes heures de début et de fin.

Précédence des cours: Il peut y avoir des situations où un cours doit précéder un autre. Dans ce cas, le cours "avant" doit être planifié de manière à se terminer avant le début du cours "après". Si les cours ont lieu le même jour, le cours "avant" doit se terminer avant le début du cours "après".

Indisponibilités des professeurs: Les indisponibilités des professeurs doivent être prises en compte lors de la planification des cours. Les cours d'un professeur ne doivent pas être planifiés pendant ses périodes d'indisponibilité.

3. Modèle prenant en compte les salles

3.1. Données

Dans cette analyse, nous utilisons les mêmes données que celles présentées dans la Section Modèle simple. Cependant, pour cette partie, nous nous concentrons sur les aspects liés aux salles. Les variables d'intérêt incluent la capacité de la salle et les services offerts par exemple.

Dans les sections suivantes, nous analyserons l'impact de ces variables sur les résultats.

3.2. Variables de décision

Dans notre modèle, nous utilisons les variables de décision suivantes pour représenter l'emploi du temps des cours et de leurs séances :

- **planning_cours_jours**: Une matrice où chaque élément (c, s) représente le jour de la semaine où la séance s du cours c est prévue. Les jours sont représentés par des entiers compris entre 0 et `nbJours`. La valeur 0 est utilisée comme valeur "poubelle" pour les séances qui ne doivent pas avoir lieu.
- **planning_cours_creneaux**: Une matrice d'intervalles où chaque élément (c, s) représente le créneau horaire de la séance s du cours c. Les intervalles sont définis dans la plage `creneaux` et ont une taille déterminée par la durée du cours c, stockée dans `creneauxSizes[c]`.
- **planning_cours_salles**: Une matrice où chaque élément (c, s) représente la salle dans laquelle se déroule la séance s du cours c. Les salles sont représentées par des entiers, avec un mappage entre les identifiants de salle et les entiers définis par `sallesInt`.

3.3. Contraintes

Contrainte de non-conflit de salle : Aucune salle ne peut être utilisée en même temps par plusieurs cours. Cette contrainte est exprimée par une double boucle `forall` sur tous les tuples de cours différents possibles et sur toutes les séances ayant lieu, et vérifie que deux cours distincts ne sont pas planifiés dans la même salle au même moment.

Contrainte de besoins et de capacité de la salle : La salle choisie doit respecter les besoins nécessaires et la taille de l'effectif pour chaque cours. Cette contrainte est exprimée par une double boucle `forall` sur tous les cours et toutes les séances ayant lieu, et vérifie que la salle choisie pour chaque séance respecte les besoins nécessaires et la capacité de la salle.

Contrainte d'indisponibilité des salles : Cette contrainte permet de gérer les indisponibilités des salles, c'est-à-dire les moments où une salle ne peut pas être utilisée. Elle est exprimée par une triple boucle `forall` sur toutes les indisponibilités de salles, tous les cours et toutes les séances ayant lieu, et vérifie que la salle choisie pour chaque séance ne correspond pas à une salle indisponible pour cette plage horaire.

4. Modèle pour un EDT compact

4.1. Données

Dans notre modèle, nous cherchons également à minimiser les trous dans l'emploi du temps. Pour cela, nous utilisons les mêmes données que celles précédemment évoquées, à savoir les informations sur les cours, les enseignants, les salles et les étudiants, ainsi que les contraintes liées à leur disponibilité.

En effet, pour minimiser les trous dans l'emploi du temps, nous devons tenir compte des créneaux horaires disponibles pour chaque cours, chaque enseignant, chaque salle et chaque étudiant, afin de les utiliser de manière optimale. Nous devons également prendre en compte les disponibilités de chaque salle pour éviter les conflits entre les cours.

En utilisant ces données, nous sommes en mesure de créer un emploi du temps optimal, avec des cours qui s'enchaînent de manière cohérente et sans trop de temps morts entre les cours. Cela permet aux étudiants et aux enseignants de mieux se concentrer et d'optimiser leur temps de travail.

4.2. Variables de décision

Dans cette section, nous nous concentrons sur la variable de décision `parcoursTrous` qui est essentielle pour optimiser la planification des cours afin d'éviter les trous dans l'emploi du temps des étudiants. Les variables `planning_cours_jours` et `planning_cours_creneaux` ont déjà été présentées précédemment.

1. `parcoursTrous` : Cette variable est dédiée à la minimisation des trous dans l'emploi du temps des étudiants pour chaque parcours. Un trou est défini comme un intervalle sans cours entre deux cours dans la même journée. La variable `parcoursTrous` prend des valeurs entières comprises entre 0 et `nbJours * nbCreneauxParJour * nbCreneauxParJour`. Cette borne supérieure est choisie pour être suffisamment large afin de pouvoir compter tous les trous possibles entre les cours.

La variable de décision `parcoursTrous` est essentielle pour l'optimisation de notre modèle, car elle nous permet de

quantifier et de minimiser les trous dans l'emploi du temps des étudiants. En minimisant cette variable, nous pouvons obtenir un emploi du temps plus compact et efficace, permettant aux étudiants de mieux gérer leur temps et d'améliorer leur expérience académique.

4.3. Contraintes

Nous avons déjà discuté des variables de décision, y compris `parcoursTrous`, qui est essentiel pour cette minimisation.

La fonction objectif du modèle vise à minimiser le nombre total de trous dans les différents parcours.

Les contraintes du modèle sont construites de manière à remplir le tableau du nombre de trous pour chaque parcours. Pour tous les cours et leurs séances, on fait la somme de l'écart entre les intervalles (trous) seulement si les deux cours appartiennent au même parcours et qu'ils ont lieu le même jour.

Pour vérifier que les deux cours appartiennent au même parcours, on utilise la condition (parcoursCours$[c1.\text{code}][p.\text{id}] = 1$ et parcoursCours$[c2.\text{code}][p.\text{id}] = 1$). Pour vérifier qu'ils ont lieu le même jour, on utilise la condition (planning_cours_jours$[c1.\text{code}][s] =$ planning_cours_jours$[c2.\text{code}][k]$).

Enfin, pour calculer le trou entre les deux cours, on utilise les informations de début et de fin des créneaux horaires des cours. Si le cours $c1$ a lieu avant le cours $c2$, on soustrait la fin du créneau horaire du cours $c1$ du début du créneau horaire du cours $c2$. Dans le cas contraire, on soustrait la fin du créneau horaire du cours $c2$ du début du créneau horaire du cours $c1$.

5. Amélioration du modèle précédent

Dans cette sous-section, nous aborderons l'amélioration du modèle précédent pour optimiser davantage la planification des emplois du temps, en utilisant les mêmes données que précédemment. En plus de minimiser le nombre de trous dans l'emploi du temps, nous chercherons également à minimiser le nombre de jours de cours et à faire en sorte que les cours se terminent plus tôt dans la journée. Ces deux améliorations permettront d'optimiser davantage la planification des emplois du temps, en offrant aux étudiants une meilleure gestion de leur temps et une expérience académique plus agréable, tout en conservant les données initiales du modèle précédent.

5.1. Variables de décisions

Dans cette partie, nous conservons les variables de décision du modèle précédent, tout en ajoutant une nouvelle variable pour améliorer l'optimisation de l'emploi du temps. La nouvelle variable, appelée `parcoursTrousJours`, sera utilisée pour compter le nombre jours entre des cours d'un même parcours/

- **parcoursTrousJours** : pour le nombre de trous de chaque parcours. Cette variable prend des valeurs entières dans l'intervalle 0 à nbJours $\times$ nbJours $\times$ nbCreneauxParJour $\times$ nbCreneauxParJour. Ce domaine large est nécessaire car nous allons compter les trous entre deux cours, même s'ils ont lieu des jours différents. P

5.2. Contraintes

Dans cette partie, nous conservons les contraintes du modèle précédent, tout en apportant des modifications à la fonction objective et en ajoutant une nouvelle contrainte pour prendre en compte les trous entre les jours de cours.

5.3. Modification de la fonction objective

La fonction objective est modifiée pour minimiser à la fois le nombre total de trous en termes de créneaux et de jours dans les différents parcours, et la somme des fins des cours pour faire en sorte que les cours se terminent plus tôt. Pour ce faire, nous utilisons un facteur de 1000 pour privilégier d'abord l'écart en jours.

5.4. Ajout de la contrainte pour les trous de jours

Une nouvelle contrainte est ajoutée pour remplir le tableau du nombre de trous en jours pour chaque parcours. Pour chaque parcours, pour tous les cours et leurs séances, nous faisons la somme des écarts entre les jours (trous) seulement si les deux cours appartiennent à ce parcours et qu'ils n'ont pas lieu le même jour.

En conservant les contraintes du modèle précédent, en modifiant la fonction objective et en ajoutant cette nouvelle contrainte pour les trous de jours, nous améliorons le modèle pour optimiser davantage la planification des emplois du temps.

5.5. Exemple de solution

Dans cette section, nous présentons un exemple de solution obtenue à partir du modèle développé. L'image illustrant l'emploi du temps généré se trouve en Annexe A.1.

Pour chaque cours, nous avons son code, le créneau horaire auquel il est programmé, le jour où il apparaît et la salle dans laquelle il se déroule. Cette solution respecte toutes les contraintes définies dans le modèle et optimise l'emploi du temps pour minimiser le nombre de jours de cours et faire en sorte que les cours se terminent plus tôt dans la journée.

Cet exemple illustre le fonctionnement du modèle et sa capacité à générer des emplois du temps optimisés en tenant compte des différentes contraintes et objectifs définis.

6. Conclusion

En conclusion, nous avons présenté et développé un modèle de création d'emploi du temps en utilisant le langage OPL. Nous avons introduit plusieurs variables de décision et contraintes pour optimiser la planification des cours. De plus, nous avons amélioré le modèle initial pour minimiser le nombre de jours de cours et faire en sorte que les cours se terminent plus tôt dans la journée.

Il serait intéressant d'explorer les performances du modèle sur des données réelles et de comparer les résultats obtenus avec d'autres méthodes de planification d'emploi du temps. Cela permettrait d'évaluer l'efficacité de notre approche et de déterminer si elle est adaptée à une utilisation dans des situations réelles.

A. Annexe

A.1. Illustration de l'emploi du temps généré

```
// solution
code : PFA
jour : 3 creneau :  <1 0 2 2> salle : PUIO-PetitAmphi
jour : 8 creneau :  <1 0 2 2> salle : PUIO-PetitAmphi
jour : 13 creneau :  <1 0 2 2> salle : PUIO-PetitAmphi
taille creneau : 2
occurence : 3
-----------
code : PFA-TP1
jour : 6 creneau :  <1 0 2 2> salle : PUIO-C202
jour : 11 creneau :  <1 0 2 2> salle : PUIO-C202
jour : 15 creneau :  <1 0 2 2> salle : PUIO-C202
taille creneau : 2
occurence : 3
-----------
code : PFA-TP2
jour : 6 creneau :  <1 0 2 2> salle : PUIO-C101
jour : 11 creneau :  <1 0 2 2> salle : PUIO-C101
jour : 15 creneau :  <1 0 2 2> salle : PUIO-C101
taille creneau : 2
occurence : 3
-----------
code : LogiqueL3
jour : 2 creneau :  <1 0 2 2> salle : PUIO-GrandAmphi
jour : 7 creneau :  <1 0 2 2> salle : PUIO-PetitAmphi
jour : 12 creneau :  <1 0 2 2> salle : PUIO-GrandAmphi
taille creneau : 2
occurence : 3
-----------
code : LogL3-TD1
jour : 4 creneau :  <1 0 2 2> salle : PUIO-C203
jour : 9 creneau :  <1 0 2 2> salle : PUIO-C203
jour : 14 creneau :  <1 0 2 2> salle : PUIO-C105
taille creneau : 2
occurence : 3
-----------
```

Figure 1. Exemple d'emploi du temps généré par le modèle