Sorbonne Université
Master ANDROIDE

Attribution multi-critère d'UE en master

UE de projet M1

Jules Mazlum - Camélia Bouali

9 mai 2025

Table des matières

Chapitre 1

Introduction

Dans le contexte universitaire, la gestion des emplois du temps représente un défi majeur, en particulier en raison de la diversité des parcours et des contraintes multiples à prendre en compte. Ce processus doit répondre à des exigences complexes : respect des horaires, gestion des capacités des groupes, prise en compte des spécificités des parcours de formation, ainsi que des préférences des étudians pour certaines Unités d'Enseignement (UE). Chaque programme possède ses propres exigences, avec des UE obligatoires et des options limitées selon les parcours. Les systèmes actuels peinent souvent à intégrer l'ensemble de ces enjeux de manière optimale.

Ce projet a pour objectif de proposer une approche plus flexible pour l'atttribution des UE aux étudiants de master, en adoptant une optimisation multi-critère. Il s'agit de générer des emplois du temps qui respectent les contraintes académiques tout en optimisant la gestion des ressources et en offrant une solution qui s'adapte aux spécificités de chaque programme.

L'approche développée repose sur la modélisation du problème sous forme de programme linéaire, qui sera résolu à l'aide du solveur Gurobi. L'algorithme conçu devra être performant pour traiter un grand nombre de données, tout en étant suffisamment flexible pour pouvoir être adapté à l'évolution des programmes chaque annéee. L'objectif est de rendre le processus d'attribution des emplois du temps plus efficace et potentiellement automatisable sur le long terme.

Le projet s'inscrit dans le cadre de l'UE de projet du Master 1 ANDROIDE de Sorbonne Université. Il est encadré par Aurélie Beynier et Olivier Spanjaard, qui assurent le suivi scientifique et méthodologique. Le code source, les données utilisées, ainsi que les scripts d'expérimentation, sont disponibles sur le dépôt GitHub suivant : https://github.com/cmla16/PROJET_ANDROIDE

Ce rapport présente la démarche suivie, les outils utilisés et les résultats obtenus dans le cadre de ce projet d'optimisation multi-critère.

Chapitre 2

État de l'art

La génération d'emplois du temps pour les étudiants est une tâche complexe relevant de l'optmisation combinatoire sous contraintes. Il s'agit de répartir des ressources limitées (salles, crénaux horaires, enseignants) entre un grand nombre d'étudiants issus de diverses formations, tout en respectant des contraintes pédagogiques (incompatibilités horaires, capacités de groupes) et des préférences individuelles. Dans ce contexte, les travaux existants s'organisent généralement autour de deux grandes approches : les modèles mono-objectifs et les modèles multi-objectifs.

2.1 Outil existant : une approche mono-objectif

L'outil actuellement utilisée par l'université pour concevoir les contrats pédagogiques et former les groupes de TD repose sur une formulation mono-objectif, dont le seul objectif est de minimiser le nombre d'étudiants sans emploi du temps valide. Autrement dit, le système cherche à attribuer un emploi du temps à un maximum d'étudiants, sans se préoccuper de la qualité de ces emplois du temps.

Cette approche présente l'avantage de la simplicité, tant en termes de modélisation que de résolution algorithmique. Cependant, elle montre rapidement ses limites : elle ignore les préférences des étudiants, ce qui peut nuire à leur satisfaction, et ne prend pas en compte des aspects qualitatifs essentiels tels que la cohérence des parcours. Il peut ainsi arriver qu'un étudiant n'obtienne pas les UE correspondant à son propre parcours, même lorsqu'il les avait placées en tête de ses choix. Cette approche peut donc générer des solutions valides mais peu satisfaisantes.

2.2 Optimisation multi-objectif et méthodes d'agrégation

Face à ces limites, une approche alternative consiste à formuler le problème comme une optimisation multi-objectif. L'idée est alors d'optimiser plusieurs objectifs simultanément, ce qui permet de mieux prendre en compte la diversité des enjeux. Dans notre cas, on va chercher à considérer les objectifs suivants :

1. Minimiser le nombre d'étudiants n'obtenant pas leur emploi du temps préféré.
2. Minimiser le nombre d'étudiants n'obtenant pas les UE de leur parcours.
3. Minimiser le nombre d'étudiants sans emploi du temps valide.

L'un des défis majeurs de l'optimisation multi-objectif est la combinaison de ces objectifs souvent conflictuels. Plusieurs stratégies d'agrégation sont possibles :

— **Somme pondérée** : on associe un poids à chaque objectif et on minimise la somme pondérée. C'est simple mais cela ne permet d'obtenir que les solutions non dominées situées sur l'enveloppe convexe du front de Pareto.

— **Ordre lexicographique** : les objectifs sont classés par priorité stricte. On optimise le premier, puis le second sans détériorer le premier, etc. Cette méthode est hiérarchique mais rigide.

— **Autres méthodes** : des approches comme le minmax regret, la méthode ϵ-contrainte sont également utilisées pour générer une diversité de solutions optimales.

2.3 Notre approche : fonction de Tchebycheff pondérée augmentée

2.3.1 Méthode de Tchebycheff pondérée augmentée classique

Dans notre projet, nous avons choisi d'utiliser une méthode d'agrégation plus avancée : la fonction de Tchebycheff pondérée augmentée. Cette méthode repose sur la minimisation de la valeur maximale des écarts pondérés entre chaque objectif et son idéal (valeur optimale isolée), accompagnée d'un terme d'augmentation linéaire afin déviter que l'optimisation ne stagne, garantissant ainsi une exploration plus complète de l'espace des solutions.

Formellement, la fonction de Tchebycheff pondérée augmentée est définie par :

$$\min \max_i \left(\lambda_i \cdot |f_i(x) - t_i^*|\right) + \varepsilon \cdot \sum_{i=1}^{n} \lambda_i \cdot |f_i(x) - t_i^*| \tag{2.1}$$

où :

— $f_i(x)$ représente la valeur de l'objectif t_i pour la solution x,

— t_i^* est la valeur idéale (la meilleure valeur atteignable) de l'objectif t_i,

— λ_i est le poids associé à l'objectif t_i,

— ε est un petit paramètre strictement positif

2.3.2 Mise à l'échelle de la fonction

Cependant, la fonction de Tchebycheff pondére augmentée présente parfois des difficultés de mise à l'échelle des écarts entre les objectifs, en particulier lorsque les objectifs ont des plages de valeurs très différentes. Pour résoudre ce problème, un terme de mise à l'échelle est utilisé, afin de garantir que tous les objectifs soient pris en compte de manière équitable dans le processus d'optimisation.

La version mise à l'échelle de la fonction peut être exprimée de la façon suivante :

$$\min \max_i \left(\lambda_i \cdot \left|\frac{f_i(x) - t_i^*}{t_i^*}\right|\right) + \varepsilon \cdot \sum_{i=1}^{n} \lambda_i \cdot \left|\frac{f_i(x) - t_i^*}{t_i^*}\right| \tag{2.2}$$

2.3.3 Justification de l'approche choisie

Cette approche présente plusieurs avantages :

— **Exploration plus large** : elle permet d'atteindre n'importe quelle solution non dominée, même celles qui sont en dehors de l'enveloppe convexe du front de Pareto, contrairement à la somme pondérée.
— **Équilibre des objectifs** : elle favorise l'équilibre entre les objectifs en évitant qu'un objectif ne domine trop fortement les autres (empêche les "effets de noyade").

Ainsi, cette méthode constitue un bon compromis entre exploration et équité entre objectifs. Nous l'avons donc retenue comme base pour notre contribution, détaillée dans le chapitre suivant.

Chapitre 3

Contribution

Dans ce chapitre, nous allons présenter la contribution principale de notre projet. Nous commencerons par formaliser les contraintes académiques et les ressources liées à l'attribution des emplois du temps des étudiants. Ensuite, nous définirons les modèles monocritères associés aux trois objectifs du projet. Nous procéderons à l'agrégation de ces objectifs en une approche multicritère, permettant ainsi d'explorer les différents compromis. Enfin, nous analyserons les résultats obtenus, afin d'évaluer l'efficacité de notre approche multi-critère.

3.1 Formalisation des contraintes académiques

Avant de définir les modèles monocritères, il est nécessaire de formaliser les contraintes qui régissent l'attribution des emplois du temps. Ces contraintes sont liées aux exigences académiques ainsi qu'aux ressources disponibles. Pour ce faire, nous allons commencer par introduire les variables de décision ainsi que les paramètres qui nous seront nécessaires pour formaliser celles-ci :

Variables :

— $x_{e,u} = 1$ si l'étudiant e est inscrit à l'UE u, 0 sinon.

— $y_{e,u,g} = 1$ si l'étudiant e est inscrit dans le groupe g de l'UE u, 0 sinon.

Paramètres :

— E : ensemble des étudiants.

— U : ensemble de toutes les UE disponibles.

— $U_e^{\text{oblig}} \subseteq U$: ensemble des UE obligatoires pour l'étudiant e.

— $U_e^{\text{parcours}} \subseteq U$: ensemble des UE proposées dans le parcours suivi par l'étudiant e

— $U_e^{\text{choix}} \subseteq U$: ensemble des UE que l'étudiant e a sélectionnées dans ses choix.

— $U_e^{\text{first}} \subseteq U$: ensemble des UE que l'étudiant e a sélectionnées dans ses premiers choix.

— ECTS_u : nombre de crédits ECTS associés à l'UE u.

— totalECTS_e : nombre de crédits ECTS que l'étudiant e doit obtenir pour valider son contrat pédagogique.

— maxHorsParcours_e : nombre maximum d'UE hors parcours autorisées pour l'étudiant e.

— IncompCM : paires d'UE dont les cours magistraux sont incompatibles (même créneau horaire).

— IncompTD : ensemble des couples $((u_1, g_1), (u_2, g_2))$ formés de deux groupes TD/TME incompatibles (se chevauchent dans l'emploi du temps), possiblement issus d'unités d'enseignement différentes.
— IncompCM_TD : ensemble des couples formés d'une UE u_1 (cours magistral) et d'un couple (u_2, g_2) (groupe TD/TME d'une autre UE) incompatibles (se chevauchent dans l'emploi du temps).

Il y a 9 types de contraintes liées à la validité des contrats pédagogiques et à la réalisabilité des emplois du temps :

1. **Contrainte sur les UE obligatoires** : Chaque étudiant doit suivre les UE obligatoires pour valider son parcours. Cela est formulé par la contrainte :

$$x_{e,u} = 1 \quad \forall u \in U_e^{\text{oblig}}, \quad \forall e \in E$$

2. **Contrainte sur la validité de l'emploi du temps** : Chaque étudiant doit être inscrit à suffisamment d'UE pour atteindre le nombre de crédits ECTS requis pour valider un contrat pédagogique.

$$\sum_{u \in U_e^{\text{choix}}} x_{e,u} \cdot \text{ECTS}_u = \text{totalECTS}_e, \quad \forall e \in E$$

3. **Contrainte d'incompatibilité entre UE :**
Pour tout $e \in E$ et pour toute paire d'UE (u_1, u_2) telles que u_1 et u_2 sont incompatibles,

$$x_{e,u_1} + x_{e,u_2} \leq 1$$

4. **Contrainte sur le nombre d'UE hors parcours autorisées :** Chaque étudiant e ne peut être affecté à plus de maxHorsParcours_e unités d'enseignement qui ne font pas partie de son parcours.

$$\sum_{u \in U \setminus U_e^{\text{parcours}}} x_{e,u} \leq \text{maxHorsParcours}_e \quad \forall e \in E$$

5. **Contrainte d'incompatibilité entre cours magistraux :** Deux unités d'enseignement u_1 et u_2 dont les cours magistraux sont incompatibles (même créneau horaire) ne peuvent pas être suivies simultanément par un même étudiant.

$$x_{e,u_1} + x_{e,u_2} \leq 1 \quad \forall e \in E, \quad \forall (u_1, u_2) \in \text{IncompCM}$$

6. **Contrainte d'incompatibilité entre TD/TME :** Deux groupes de TD ou TME incompatibles (même créneau horaire), appartenant à potentiellement deux unités d'enseignement différentes, ne peuvent pas être suivis simultanément par un même étudiant.

$$y_{e,u_1,g_1} + y_{e,u_2,g_2} \leq 1 \quad \forall e \in E, \quad \forall ((u_1, g_1), (u_2, g_2)) \in \text{IncompTD}$$

7. **Contrainte d'incompatibilité entre TD/TME et CM :** Un étudiant ne peut pas être inscrit à un cours magistral et à un TD/TME qui ont lieu en même temps.

$$x_{e,u_1} + y_{e,u_2,g_2} \leq 1 \quad \forall e \in E, \quad \forall (u_1, (u_2, g_2)) \in \text{IncompCM\_TD}$$

8. **Contrainte d'unicité de groupe par UE :** Un étudiant ne peut être inscrit qu'à un seul groupe de TD/TME pour une même unité d'enseignement.

$$\sum_{g \in G_u} y_{e,u,g} = x_{e,u} \quad \forall e \in E, \quad \forall u \in U$$

9. **Contrainte de capacité des groupes de TD/TME :**

$$\sum_{e \in E} y_{e,u,g} \leq \text{cap}_{u,g} \quad \forall u \in U, \quad \forall g \in G_u$$

où $\text{cap}_{u,g}$ est la capacité maximale du groupe g de l'UE u et G_u est l'ensemble des groupes de l'UE u.

3.2 Formalisation des modèles monocritères

Dans cette section, nous allons formaliser les modèles monocritères pour chacun des objectifs définis précédemment. Chaque objectif sera formulé en tant que problème d'optimisation, où l'on cherchera à minimiser une fonction objectif sous les contraintes formulées dans la section précédente.

3.2.1 Objectif 1 : Minimisation du nombre d'étudiants n'obtenant pas leur emploi du temps préféré

L'objectif est de réduire le nombre d'étudiants qui ne sont pas inscrits dans les unités d'enseignement qu'ils ont classées en priorité. Pour cela, nous allons introduire une nouvelle variable de décision :

$$z_e^1 = \begin{cases} 1 & \text{si l'étudiant } e \text{ n'est pas inscrit à au moins une de ses UE préférées} \\ 0 & \text{sinon} \end{cases} \quad \forall e \in E$$

L'objectif devient alors :

$$\min f_1 = \sum_{e \in E} z_e^1$$

$$\begin{cases} z_e^1 \geq 1 - \frac{1}{|U_e^{\text{first}}|} \sum_{u \in U_e^{\text{first}}} x_{e,u} & \forall e \in E \\ x_{e,u} = 1 & \forall u \in U_e^{\text{oblig}},\ \forall e \in E \\ \sum_{u \in U_e^{\text{choix}}} x_{e,u} \cdot \text{ECTS}_u = \text{totalECTS}_e & \forall e \in E \\ x_{e,u_1} + x_{e,u_2} \leq 1 & \forall e \in E,\ \forall (u_1, u_2) \text{ incompatibles} \\ \sum_{u \in U \setminus U_e^{\text{parcours}}} x_{e,u} \leq \text{maxHorsParcours}_e & \forall e \in E \\ x_{e,u_1} + x_{e,u_2} \leq 1 & \forall e \in E,\ \forall (u_1, u_2) \in \text{IncompCM} \\ y_{e,u_1,g_1} + y_{e,u_2,g_2} \leq 1 & \forall e \in E,\ \forall ((u_1, g_1), (u_2, g_2)) \in \text{IncompTD} \\ x_{e,u_1} + y_{e,u_2,g_2} \leq 1 & \forall e \in E,\ \forall (u_1, (u_2, g_2)) \in \text{IncompCM\_TD} \\ \sum_{g \in G_u} y_{e,u,g} = x_{e,u} & \forall e \in E,\ \forall u \in U \\ \sum_{e \in E} y_{e,u,g} \leq \text{cap}_{u,g} & \forall u \in U,\ \forall g \in G_u \end{cases}$$

3.2.2 Objectif 2 : Minimisation du nombre d'étudiants n'obtenant pas les UE de leur parcours

Cet objectif vise à faire en sorte que chaque étudiant puisse être affecté à un maximum d'unités d'enseignement faisant partie de son parcours académique. On introduit une nouvelle variable de décision :

$$z_e^2 = \begin{cases} 1 & \text{si l'étudiant } e \text{ n'a pas au moins une des ses UE de parcours} \\ 0 & \text{sinon} \end{cases} \quad \forall e \in E$$

L'objectif devient alors :

$$\min f_2 = \sum_{e \in E} z_e^2$$

$$\begin{cases}
z_e^2 \geq 1 - \frac{1}{|U_e^{\text{parcours}} \cap U_e^{\text{first}}|} \cdot \sum_{u \in U_e^{\text{parcours}} \cap U_e^{\text{first}}} x_{e,u} & \forall e \in E \\
x_{e,u} = 1 & \forall u \in U_e^{\text{oblig}},\ \forall e \in E \\
\sum_{u \in U_e^{\text{choix}}} x_{e,u} \cdot \text{ECTS}_u = \text{totalECTS}_e & \forall e \in E \\
x_{e,u_1} + x_{e,u_2} \leq 1 & \forall e \in E,\ \forall (u_1, u_2) \text{ incompatibles} \\
\sum_{u \in U \setminus U_e^{\text{parcours}}} x_{e,u} \leq \text{maxHorsParcours}_e & \forall e \in E \\
x_{e,u_1} + x_{e,u_2} \leq 1 & \forall e \in E,\ \forall (u_1, u_2) \in \text{IncompCM} \\
y_{e,u_1,g_1} + y_{e,u_2,g_2} \leq 1 & \forall e \in E,\ \forall ((u_1, g_1), (u_2, g_2)) \in \text{IncompTD} \\
x_{e,u_1} + y_{e,u_2,g_2} \leq 1 & \forall e \in E,\ \forall (u_1, (u_2, g_2)) \in \text{IncompCM_TD} \\
\sum_{g \in G_u} y_{e,u,g} = x_{e,u} & \forall e \in E,\ \forall u \in U \\
\sum_{e \in E} y_{e,u,g} \leq \text{cap}_{u,g} & \forall u \in U,\ \forall g \in G_u
\end{cases}$$

3.2.3 Objectif 3 : Minimisation du nombre d'étudiants sans emploi du temps valide

Enfin, le dernier objectif cherche à limiter le nombre d'étudiants dont l'emploi du temps ne satisfait pas les contraintes institutionnelles. Pour cela, on doit réadapter la contrainte sur la validité d'un emploi du temps : on introduit une variable d'écart ec_e qui mesure le nombre d'ECTS manquant pour obtenir un emploi du temps valide. De plus, on introduit la variable de décision suivante :

$$z_e^3 = \begin{cases} 1 & \text{si l'étudiant } e \text{ n'a pas d'emploi du temps valide} \\ 0 & \text{sinon} \end{cases} \qquad \forall e \in E$$

L'objectif devient alors :

$$\min f_3 = \sum_{e \in E} z_e^3$$

$$\begin{cases}
\sum_{u \in U_e^{\text{choix}}} x_{e,u} \cdot \text{ECTS}_u + ec_e = \text{totalECTS}_e & \forall e \in E \\
ec_e \leq 30 \cdot z_e^3 & \forall e \in E \\
x_{e,u} = 1 & \forall u \in U_e^{\text{oblig}},\ \forall e \in E \\
x_{e,u_1} + x_{e,u_2} \leq 1 & \forall e \in E,\ \forall (u_1, u_2) \text{ incompatibles} \\
\sum_{u \in U \setminus U_e^{\text{parcours}}} x_{e,u} \leq \text{maxHorsParcours}_e & \forall e \in E \\
x_{e,u_1} + x_{e,u_2} \leq 1 & \forall e \in E,\ \forall (u_1, u_2) \in \text{IncompCM} \\
y_{e,u_1,g_1} + y_{e,u_2,g_2} \leq 1 & \forall e \in E,\ \forall ((u_1, g_1), (u_2, g_2)) \in \text{IncompTD} \\
x_{e,u_1} + y_{e,u_2,g_2} \leq 1 & \forall e \in E,\ \forall (u_1, (u_2, g_2)) \in \text{IncompCM_TD} \\
\sum_{g \in G_u} y_{e,u,g} = x_{e,u} & \forall e \in E,\ \forall u \in U \\
\sum_{e \in E} y_{e,u,g} \leq \text{cap}_{u,g} & \forall u \in U,\ \forall g \in G_u
\end{cases}$$

3.3 Agrégation des objectifs en approche multi-critère

Nous avons choisi d'agréger les trois objectifs décrits précédemment à l'aide de la fonction de Tchebycheff pondérée augmentée, présentée dans la section 2.3. Cette méthode nous permet de rechercher une solution qui équilibre les trois critères tout en assurant une certaine équité entre eux.

Dans notre cas, les trois fonctions objectifs à minimiser sont les suivantes :

$$\begin{aligned}
f_1(x) &= \text{nombre d'étudiants n'obtenant pas leur emploi du temps préféré,}\\
f_2(x) &= \text{nombre d'étudiants n'obtenant pas les UE de leur parcours,}\\
f_3(x) &= \text{nombre d'étudiants sans emploi du temps valide.}
\end{aligned}$$

L'agrégation de ces trois objectifs est formulée de la manière suivante :

$$\min \max_{i=1,2,3} \left(\lambda_i \cdot \left| \frac{f_i(x) - t_i^*}{t_i^*} \right| \right) + \varepsilon \cdot \sum_{i=1}^{3} \lambda_i \cdot \left| \frac{f_i(x) - t_i^*}{t_i^*} \right| \tag{3.1}$$

où :
- t_i^* : meilleure valeur atteinte pour f_i.
- λ_i est le poids affecté à l'objectif f_i, représentant son importance relative.
- ε est un petit paramètre strictement positif permettant d'éviter les solutions de stagnation.

Pour linéariser ce programme, on introduit une variable z telle que $z = \max_{i=1,2,3} \left(\lambda_i \cdot \left| \frac{f_i(x)-t_i^*}{t_i^*} \right| \right)$. De plus, on pose $e_i = \left| \frac{f_i(x)-t_i^*}{t_i^*} \right|$. La fonction objectif devient alors :

$$\min f = z + \varepsilon \cdot \sum_{i=1}^{3} \lambda_i \cdot e_i \tag{3.2}$$

$$\begin{cases}
z \geq \lambda_1 \cdot e_1 & \\
z \geq \lambda_2 \cdot e_2 & \\
z \geq \lambda_3 \cdot e_3 & \\
z_e^1 \geq 1 - \dfrac{1}{|U_e^{\text{first}}|} \sum_{u \in U_e^{\text{first}}} x_{e,u} & \forall e \in E \\
z_e^2 \geq 1 - \frac{1}{|U_e^{\text{parcours}} \cap U_e^{\text{first}}|} \cdot \sum_{u \in U_e^{\text{parcours}} \cap U_e^{\text{first}}} x_{e,u} & \forall e \in E \\
\sum_{u \in U_e^{\text{choix}}} x_{e,u} \cdot \text{ECTS}_u + ec_e = \text{totalECTS}_e & \forall e \in E \\
ec_e \leq 30 \cdot z_e^3 & \forall e \in E \\
x_{e,u} = 1 & \forall u \in U_e^{\text{oblig}},\ \forall e \in E \\
x_{e,u_1} + x_{e,u_2} \leq 1 & \forall e \in E,\ \forall (u_1,u_2) \text{ incompatibles} \\
\sum_{u \in U \setminus U_e^{\text{parcours}}} x_{e,u} \leq \text{maxHorsParcours}_e & \forall e \in E \\
x_{e,u_1} + x_{e,u_2} \leq 1 & \forall e \in E,\ \forall (u_1,u_2) \in \text{IncompCM} \\
y_{e,u_1,g_1} + y_{e,u_2,g_2} \leq 1 & \forall e \in E,\ \forall ((u_1,g_1),(u_2,g_2)) \in \text{IncompTD} \\
x_{e,u_1} + y_{e,u_2,g_2} \leq 1 & \forall e \in E,\ \forall (u_1,(u_2,g_2)) \in \text{IncompCM_TD} \\
\sum_{g \in G_u} y_{e,u,g} = x_{e,u} & \forall e \in E,\ \forall u \in U \\
\sum_{e \in E} y_{e,u,g} \leq \text{cap}_{u,g} & \forall u \in U,\ \forall g \in G_u
\end{cases}$$

3.4 Exploration des compromis et analyse des résultats

Afin d'explorer l'espace des solutions non dominées et de mieux comprendre les compromis entre objectifs conflictuels, nous avons mis en œuvre plusieurs stratégies de relâchement.

3.4.1 Variation des poids λ_i

Dans cette partie, nous étudions l'impact de la **pondération des objectifs** , définie dans la fonction de Tchebycheff pondérée augmentée. Les pondérations λ_i sont des coefficients appliqués à chaque objectif pour refléter leur importance relative dans l'optimisation.

itération	w_1	w_2	w_3	f_1	f_2	f_3
0	0.8	0.1	0.1	55	12	7
1	0.1	0.8	0.1	57	7	8
2	0.1	0.1	0.8	67	18	3
3	0.6	0.3	0.1	57	7	8
4	0.1	0.6	0.3	64	15	4
5	0.3	0.1	0.6	67	18	3
6	0.5	0.25	0.25	67	18	3
7	0.25	0.5	0.25	64	15	4
8	0.25	0.25	0.5	67	18	3
9	0.33	0.33	0.33	67	18	3

TABLE 3.1 – Résultats obtenus pour différentes combinaisons de poids (w_1, w_2, w_3) et les valeurs associées des fonctions objectifs (f_1, f_2, f_3).

L'analyse montre que la variation des poids permet de générer plusieurs compromis intéressants entre les objectifs. Par exemple :

— Donner un poids élevé à f_1 (itération 0) permet de fortement réduire le nombre d'étudiants insatisfaits (55), mais au prix de davantage d'erreurs sur les autres critères.

— À l'inverse, une pondération élevée sur f_3 (itérations 2, 5) permet de minimiser les erreurs de validité (3), mais entraîne un nombre plus important d'insatisfactions (67) et d'erreurs de parcours (18).

— Les pondérations intermédiaires (ex. itération 4 ou 7) offrent des compromis équilibrés entre les objectifs, sans exceller ni échouer complètement sur un critère.

Ces résultats sont également représentés dans la figure 3.1, qui affiche les points (f_1, f_2, f_3) dans l'espace tridimensionnel des objectifs.

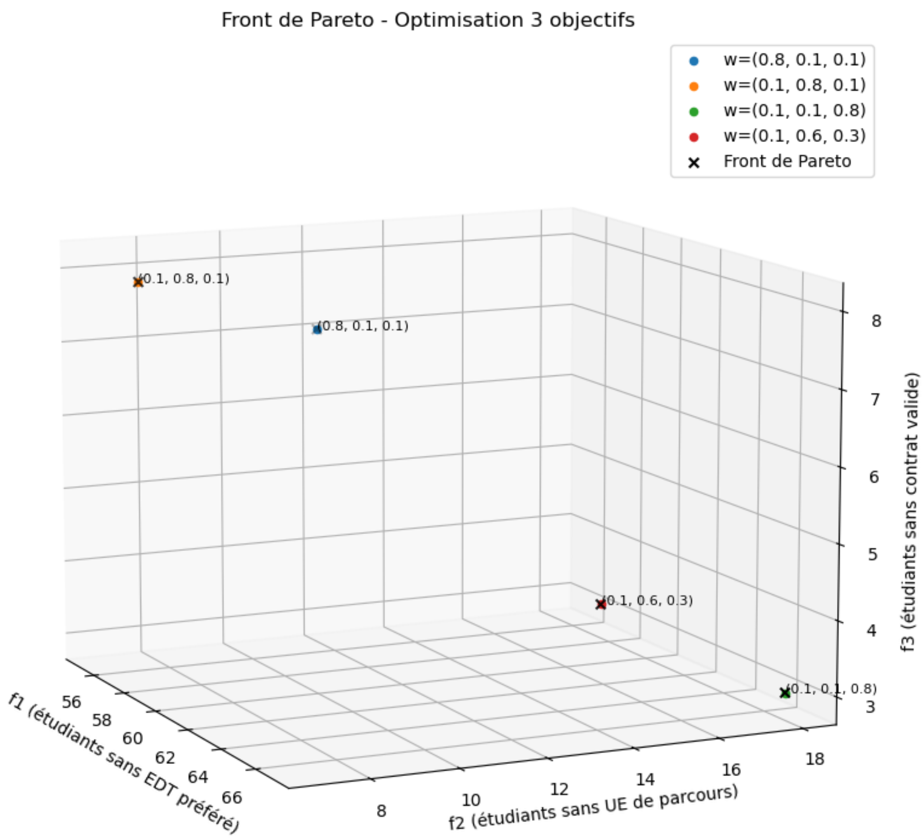

FIGURE 3.1 – Solutions obtenues pour différentes pondérations (w_1, w_2, w_3) dans l'espace des objectifs (f_1, f_2, f_3).

L'analyse des résultats montre que chaque solution représente un compromis valide entre les différents objectifs, sans qu'une solution soit objectivement meilleure que les autres sur tous les critères. La variation des poids λ_i permet d'explorer diverses solutions du front de Pareto, toutes non dominées, offrant ainsi aux décideurs plusieurs options équilibrées selon les priorités définies.

3.4.2 Relâchement des capacités des groupes

Dans cette sous-section, nous explorons l'effet du relâchement des capacités des groupes, c'est-à-dire assouplir certaines contraintes sur les groupes d'étudiants.

On a introduit une contrainte relaĉhement qui nous permet d'augmenter tous les groupes de TD/TME complets d'un petit paramètre δ.

On a fait varier le paramètre δ pour voir l'évalution des solutions. On reprèsente les différents résultats dans la table suivante :

δ	f_1	f_2	f_3
0	71.0	26.0	4.0
1	61.0	15.0	3.0
2	61.0	15.0	3.0
3	61.0	14.0	3.0
4	61.0	14.0	3.0
6	61.0	11.0	3.0
7	59.0	10.0	3.0

TABLE 3.2 – Valeurs des objectifs f_1, f_2 et f_3 selon le relâchement de la capacité maximale des groupes (δ)

On représente également ces résultats dans la figure 3.2, ci-dessous :

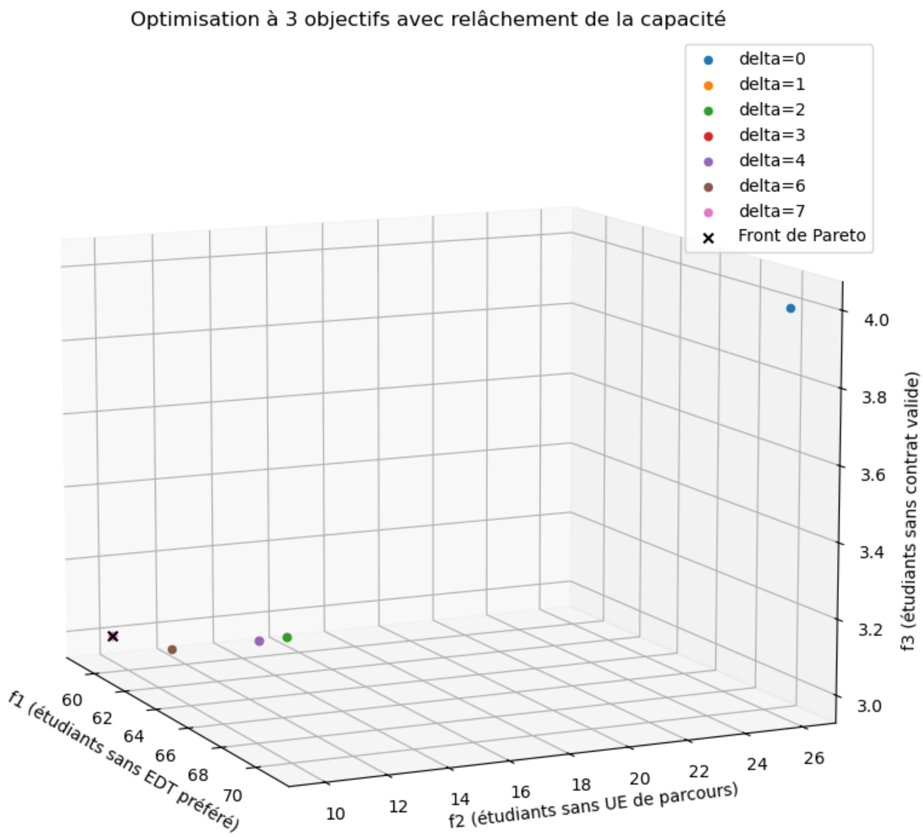

FIGURE 3.2 – Optimisation à 3 objectifs avec relâchement de la capacité : représentation de toutes les solutions avec une croix sur celles qui sont non dominées.

On peut voir que l'augmentation progressive de la capacité des groupes améliore nettement la qualité des solutions obtenues. Même un relâchement minimal permet d'atteindre un bien meilleur compromis entre les différents objectifs. Autoriser une légère flexibilité dans la taille des groupes constitue un levier simple et efficace pour améliorer l'équité et la satisfaction dans l'attribution des emplois du temps.

3.4.3 Encadrement d'un objectif par contrainte

Cette approche consiste à encadrer un objectif (par exemple f_1) par une contrainte spécifique, afin de garantir qu'il soit optimisée tout en respectant certaines limites sur d'autres objectifs.

Dans notre cas on décide de fixer l'écart pour l'objectif f_1. On commence par résoudre le programme linéaire une première fois pour obtenir la valeur optimale e_1^* sans relâcher de contraintes.

Une fois la valeur e_1^* calculée, on ajoute la contrainte suivante dans le programme linéaire :

$$e_1 \leq e_1^* - \text{écart}$$

Analyse de l'encadrement de f_1

Le graphe ci-dessous montre l'évolution des objectifs f_2 (étudiants sans UE de parcours) et f_3 (étudiants sans contrat validé) en fonction des écarts fixés sur f_1. On observe que lorsque l'on impose une contrainte plus stricte sur f_1 (écart faible), cela tend à améliorer f_2 (diminution du nombre d'étudiants sans UE de parcours), mais cela se fait au détriment de f_3, qui se dégrade (augmentation du nombre d'étudiants sans contrat validé).

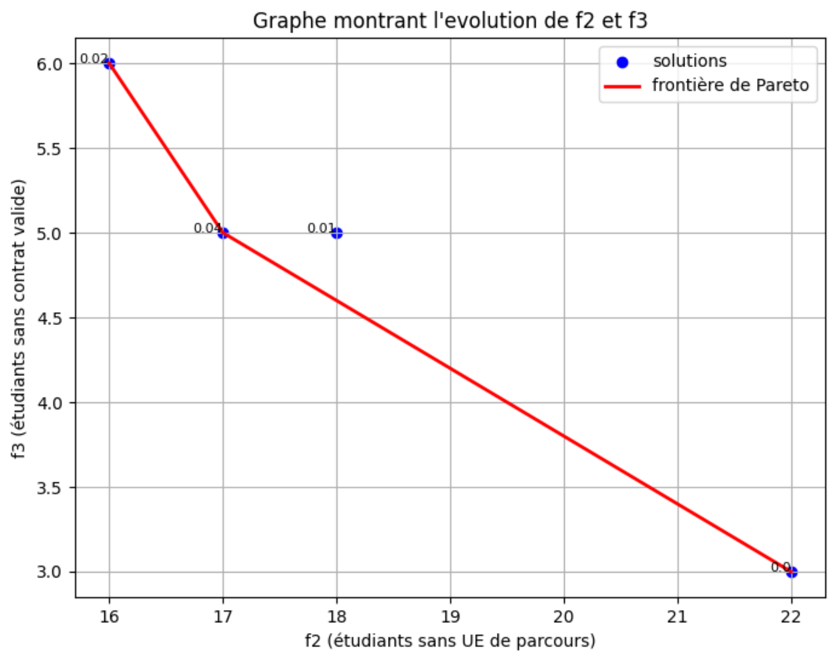

FIGURE 3.3 – Graphe d'évolution de f2 et f3 en fonction des écarts imposés à f1

Il est aussi notable que le point correspondant à un écart de 0.01 est **dominé** : il n'appartient pas à la frontière de Pareto, car une autre solution permet de faire mieux ou aussi bien sur les deux objectifs f_2 et f_3. Cela signifie que ce point n'est pas optimal au sens de Pareto, et qu'il pourrait être évité dans une stratégie de compromis entre les objectifs.

En résumé, l'encadrement de f_1 a un effet direct sur la position des solutions dans l'espace des objectifs secondaires. Il permet d'orienter la recherche vers des compromis spécifiques, mais peut aussi conduire à des solutions dominées si les contraintes sont trop strictes.

Chapitre 4

Conclusion

Dans ce projet, nous avons proposé une approche d'optimisation multi-objectif pour l'attribution des emplois du temps des étudiants de master, basée sur un modèle en programmation linéaire et la fonction de Tchebycheff pondérée augmentée. Cette méthode a permis de prendre en compte différents critères, tout en obtenant des résultats satisfaisants.

Cependant, plusieurs axes d'amélioration sont envisageables. Un point important concerne la prise en compte de l'ordre des préférences exprimées par les étudiants. Actuellement, notre modèle considère les unités d'enseignement demandées sans distinguer si elles sont classées en premier ou en deuxième par un étudiant. Cela peut engendrer une insatisfaction, notamment pour ceux dont les premiers choix ne sont pas respectés. Intégrer cette hiérarchie dans le modèle permettrait d'affiner la notion de satisfaction individuelle et de produire des emplois du temps plus proches des attentes réelles des étudiants.

Ces perspectives ouvrent la voie à des améliorations significatives, tant sur la qualité des solutions que sur leur adéquation aux besoins des étudiants et des institutions. Notre travail constitue ainsi une première étape vers un système d'attribution plus équitable, plus souple et mieux adapté aux contraintes réelles.

Annexe A

Cahier des charges

A.1 Contexte

Actuellement, l'attribution des emplois du temps des étudiants en master repose sur un modèle d'optimisation monocritère, qui minimise uniquement le nombre d'étudiants sans emploi du temps valide. Cependant, cette approche est insuffisante, car elle ne prend pas en compte d'autres aspects essentiels comme les spécificités de chaque parcours de master.

Ainsi, pour améliorer la qualité des affectations et répondre aux attentes des étudiants, nous souhaitons introduire plusieurs objectifs et adopter une approche d'optimisation multi-critère. Pour cela, nous allons développer un algorithme utilisant un solveur d'optimisation afin de traiter ces objectifs de manière équilibrée.

A.2 Données et contraintes

Notre système repose sur l'exploitation de trois documents essentiels :

1. Document des choix étudiants : Contient le numéro étudiant, son parcours, ses UE obligatoires, et ses UE classées par ordre de préférence.
2. Document des UE : Liste toutes les UE avec leur nombre de crédits ECTS, le nombre de groupes de TD/TME, la capacité de chaque groupe, ainsi que leurs horaires.
3. Document des UE de parcours : Indique les UE propres à chaque parcours.
4. Document des règles de parcours : Pour chaque parcours, indique le nombre d'UE hors parcours autorisé.
5. Document des incompatibilités : Liste les paires d'UE qui sont incompatibles entre elles.

Ces données seront extraites et formatées à l'aide d'un script Python afin d'être exploitables par notre algorithme.

Parcours de master

Le master est composé de 8 parcours distincts hors alternance :

1. ANDROIDE
2. CCA
3. DAC

4. IMA
5. QI
6. SAR
7. SESI
8. STL

Chaque parcours possède des contraintes spécifiques, notamment des UE obligatoires et un choix restreint d'UE optionnelles. Les étudiants doivent exprimer des vœux en classant leurs UE préférées, ce qui constitue une base pour l'attribution des emplois du temps.

A.3 Problématique

L'objectif est de répartir de manière optimale les étudiants dans les différents groupes de TD/TME tout en respectant plusieurs contraintes :

1. Chaque étudiant doit avoir suffisamment d'UE pour valider 30 ECTS (ou moins s'il est redoublant).
2. Chaque étudiant doit avoir les UE obligatoires de son parcours.
3. Il faut gérer les incompatibilités d'emploi du temps entre UE.
4. Il faut prendre en compte les capacités des groupes.

La question centrale est donc : comment attribuer les emplois du temps en conciliant ces contraintes académiques et les ressources disponibles tout en assurant une solution optimisée ?

A.4 Objectifs du projet

Objectif global du projet

L'objectif principal du projet est de développer un algorithme permettant d'utiliser un solveur d'optimisation afin d'attribuer les UE aux étudiants en respectant les trois objectifs du programme linéaire avec une approche multi-critère.

Objectifs du programme linéaire

Le programme linéaire que nous allons concevoir devra prendre en compte les trois objectifs suivants :

1. Minimiser la somme des rangs des préférences des étudiants.
2. Minimiser le nombre d'UE hors parcours refusé pour chaque étudiant.
3. Minimiser le nombre d'étudiants sans emploi du temps valide.

L'approche suivie consistera d'abord à formuler ces objectifs en tant que problèmes d'optimisation monocritère distincts. Ensuite, nous introduirons une approche multi-critère pour combiner ces objectifs et explorer différents compromis à travers des méthodes adaptées. Une interface de visualisation des solutions optimales sera développée si le temps le permet.

A.5 Méthode et outils

Solveur et modélisation

L'optimisation sera réalisée à l'aide du solveur Gurobi, qui est spécialisé dans la résolution de problèmes d'optimisation linéaire. Un script Python sera utilisé pour :

1. Extraire et prétraiter les données des trois documents nécessaires.
2. Modéliser le problème sous forme de programme linéaire.
3. Soumettre ce modèle au solveur Gurobi et analyser les solutions obtenues.

Interface de visualisation

Si le temps le permet, une interface de visualisation des solutions optimales sera développée afin de faciliter l'analyse des résultats.

A.6 Exigences

Exigences de performance : Le système devra être capable de traiter un grand nombre d'étudiants et d'UE en un temps raisonnable. Il est donc important que l'algorithme d'optimisation soit performant et capable de traiter un grand nombre de données sans compromettre l'efficacité.

Exigences d'adaptabilité : Le système devra être conçu de manière à pouvoir s'adapter aux variations des programmes académiques, qu'il s'agisse des différences d'UE entre les semestres ou des évolutions des maquettes pédagogiques d'une année à une autre. Ces ajustements devront être possibles sans changements majeurs dans le fonctionnement du système, pour le rendre plus facile à maintenir et à faire évoluer.

Annexe B

Manuel utilisateur

B.1 Exécution du programme

L'exécution du programme se fait depuis le répertoire `src/` du projet. Aucun module externe n'est requis, le script peut donc être lancé directement avec l'interpréteur Python.

Le script principal à exécuter est `multi123_minmax_lineaire.py`. Pour le lancer, il suffit d'ouvrir un terminal, de se placer dans le dossier `src/`, puis d'exécuter la commande suivante :

```
python multi123_minmax_lineaire.py
```

La fonction d'optimisation attend cinq fichiers en paramètres, correspondant aux différentes données nécessaires au traitement, ainsi qu'un paramètre ε et trois coefficients λ_1, λ_2 et λ_3. Ces fichiers doivent être placés dans le dossier `data/`. Il est important que leur format respecte les spécifications attendues.

À l'issue de l'exécution, deux types de résultats sont générés :

— les statistiques, qui sont enregistrées dans le dossier `output/statistiques/` ;

— les attributions, disponibles dans `output/attributions/multi123_minmax_lineaire`.

B.1.1 Options et paramètres supplémentaires

Activation d'une tolérance sur l'écart à l'optimum

Le programme permet également de relâcher certaines contraintes d'optimisation, ce qui peut s'avérer utile pour explorer des compromis entre plusieurs objectifs. Cette fonctionnalité s'active en passant des arguments supplémentaires au moment de l'exécution.

Plus précisément, il est possible d'ajouter les paramètres suivants :

— `relax_obj` : correspond à l'objectif que l'on souhaite relâcher. Il doit être donné sous forme de chaîne de caractères parmi `z1`, `z2` ou `z3`.

— `mode` : indique le mode d'exécution. Il peut prendre deux valeurs :

  — `collect` : utilisé lors de la première exécution pour collecter la valeur de référence de l'objectif choisi.

  — `relax` : utilisé lors de la seconde exécution pour autoriser un relâchement sur cet objectif.

— `gap` : écart autorisé par rapport à la valeur optimale de l'objectif relâché. Ce paramètre est requis uniquement lorsque le mode est `relax`.

L'utilisation typique se fait en deux étapes :

1. Une **première exécution** avec `mode = collect`, afin d'évaluer la valeur de l'objectif sans relâchement ;
2. Une **seconde exécution** avec `mode = relax` et un `gap` spécifié, pour autoriser une marge sur l'objectif sélectionné.

Relâchement de la contrainte sur les capacités de groupe

Afin de relâcher la contrainte sur les capacités de groupe il faut modifier les valeurs des paramètres suivants :

- `capacity` : est mis à `True` si on souhaite relâcher la contrainte et `False` sinon.
- `delta` : si le paramètre `capacity` vaut `True` alors, tous les groupes complets sont augmentés d'une petite valeur `delta`.

Ces paramètres permettent ainsi de guider la recherche vers des solutions plus équilibrées en tolérant une légère dégradation contrôlée de certaines fonctions objectif.

Bibliographie

[1] Perny, Patrice and Grabisch, Michel. *Agrégation Multicritère*. 2003. Disponible en ligne : https://www.researchgate.net/profile/Patrice-Perny/publication/267421473_Agregation_Multicritere/links/547590610cf2778985aedade/Agregation-Multicritere.pdf