



Faculté des Sciences

Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option : Modèle intelligents et décision (M.I.D)

Thème

Gestion Intelligente des Emplois du Temps :
Une Approche d'Optimisation grâce au
Génie Logiciel et à l'Intelligence Artificielle

Réalisé par : Karoun Fadia Mouna épouse Sari Hassoun

Présenté le 24 juin 2024 devant le jury composé de :

- M. Smahi Mohammed Ismail : Encadrant
- M. Belabed Amine : Président
- Mme. Marouf Radja : Examineur

Année universitaire : 2023-2024

Dédicace

Je dédie ce modeste travail à

Mes parents

Mon mari

Ma fille Manel

Mes frères

Mes sœurs

Mes amis

Toute ma famille et ma belle famille

Je remercie tous ceux et celles qui

m'ont aidé à réaliser ce travail.

Remerciements

Nous remercions le bon dieu de nous avoir mis Sur la voie
du savoir.

Nous tenons à remercier vivement notre Promoteur
M.SMAHI MOHAMMED ISMAIL Pour la qualité de son
encadrement et son Suivi durant toute la durée du projet.

Nous remercions les membres du jury pour l'honneur qu'ils
nous font en acceptant de juger Ce modeste travail.

Enfin, nous remercions toutes les personnes Ayant
contribué de près ou de loin au bon Accomplissement de
notre travail.

Résumé

Ce mémoire explore la gestion intelligente des emplois du temps des examens en utilisant une approche d'optimisation combinant le génie logiciel et l'intelligence artificielle, avec un accent sur l'Algorithme Intelligent Water Drops (IWD). L'objectif est de développer un système capable de générer des emplois du temps optimisés, minimisant les conflits et maximisant l'utilisation des ressources académiques. L'algorithme IWD, inspiré par le comportement naturel des rivières d'eau trouvant des chemins optimaux, est adapté pour résoudre le problème complexe de planification des examens. Les résultats montrent une réduction significative des conflits d'horaires et une meilleure allocation des ressources, améliorant la satisfaction des étudiants et des enseignants. Ce système intelligent s'avère être une solution efficace et adaptable aux besoins dynamiques des institutions académiques.

Abstract

This dissertation explores the intelligent management of exam timetables using an optimization approach combining software engineering and artificial intelligence, with a focus on the Intelligent Water Drops (IWD) Algorithm. The aim is to develop a system capable of generating optimized timetables, minimizing conflicts and maximizing the use of academic resources. The IWD algorithm, inspired by the natural behavior of water drops finding optimal paths, is adapted to solve the complex problem of exam scheduling. The results show a significant reduction in scheduling conflicts and a better allocation of resources, improving student and teacher satisfaction. This intelligent system is proving to be an effective and adaptable solution to the dynamic needs of academic institutions.

ملخص

تستكشف هذه الأطروحة الإدارة الذكية لجدول الامتحانات باستخدام نهج التحسين الذي يجمع بين هندسة البرمجيات والذكاء الاصطناعي، مع التركيز على خوارزمية قطرات الماء الذكية. الهدف هو تطوير نظام قادر على إنشاء جداول زمنية محسنة، وتقليل الصراعات، وتعظيم استخدام الموارد الأكاديمية. إن هذه الخوارزمية، المستوحاة من السلوك الطبيعي لأنهار المياه التي تجد المسارات المثلى، مناسبة لحل مشكلة جدولة الامتحانات المعقدة. تظهر النتائج انخفاضًا كبيرًا في تعارض المواعيد وتخصيص الموارد بشكل أفضل، مما يؤدي إلى تحسين رضا الطلاب والمعلمين. يثبت هذا النظام الذكي أنه حل فعال وقابل للتكيف مع الاحتياجات الدينامية للمؤسسات الأكاديمية.

Table Des Matières

Dédicace	i
Remerciements	ii
Résumé	iii
Table Des Matières	iv
Liste des figures	vii
Introduction Générale	1
Contexte général	1
Organisation du mémoire	1
Chapitre I : Méthodes Pour La Gestion Intelligente Des Emplois Du Temps	
1. Introduction.....	4
2. Les algorithmes d'ordonnancement	4
1. . Ordonnancement basé sur les règles.....	5
2. Algorithme de Johnson	6
3. Algorithme de séquence critique	7
4. Programmation linéaire/entière	8
3. L'apprentissage automatique	9
1. L'apprentissage supervisé	10
2. L'apprentissage non supervisé.....	10
3. L'apprentissage par renforcement	11
4. Systems experts	11
4.1. Comment fonctionne un système expert pour la gestion des emplois du temps	12
5. Optimisation multi-objectif.....	13
1. Les principaux objectifs à optimiser dans la gestion des emplois du temps	13
2. Les différentes approches d'optimisation multi objectifs	13
6. Agents intelligents	14

1. Les catégories de l'agents intelligents	14
2. Les avantages des agents intelligents pour la gestion des emplois du temps....	15
3. Les inconvénients des agents intelligents pour la gestion des emplois du temps	15
7. Les méthodes heuristiques	16
1. Les algorithmes gloutons.....	16
2. Les méthodes de recherche locale	16
3. Les algorithmes génétiques	17
4. Les méthodes hybrides	17
8. Conclusion	19

Chapitre II : Algorithme IWD pour le problème de calendrier d'examen

1 Introduction.....	21
2 Généralités sur les emplois du temps	21
1. Définition d'emplois du temps.....	21
2. Historique.....	21
3. A Quoi sert un emploi du temps.....	22
4. Comment un emploi du temps est élaboré.....	23
5. Les emplois du temps des examens universitaires	23
3. Utilisation de l'algorithme Intelligent Water Drops (IWD)	24
1. Définition de l'algorithme IWD	24
2. Application de l'algorithme IWD	24
3. Performance de l'algorithme IWD	24
4. Hybrides algorithme	26
1. Définition de l'algorithme hybrides	26
2. Algorithme Hybride IWD proposé pour PPEU	26
i. IWD pour le problème de l'horaire d'examen sans capacité	26
ii. Optimiseur de recherche locale (LSO)	32
iii. IWD hybride pour PPEU	36
3. Importance des algorithme hybrides.....	37

4. Exemples et application dans divers domaines	37
5. conclusion	39

Chapitre III: *Expérimentations Et Résultats*

1. La base de données	41
1. Les métriques utilisées	42
2. Les paramètres	43
3. Les hypers paramètres	43
2. les résultats d`expérimentation	44
3. Une discussion.....	51
Conclusion générale.....	53
Références bibliographiques	55

Liste des figures

Figure 1 : Répartition des Créneaux Horaires pour les Examens - université ear-f-83-2.....	44
Figure 2 : Répartition des Créneaux Horaires pour les Examens - université yor-f-83-2	46
Figure 3 : Répartition des Créneaux Horaires pour les Examens - Université sta-f-83-2	47
Figure 4 : Répartition des Créneaux Horaires pour les Examens - Université hec-s-92-2	48
Figure 5: pénalité pour chaque Université	50

Introduction Générale

Introduction Générale

Contexte général

La gestion des emplois du temps, en particulier pour les examens académiques, représente un défi majeur pour les institutions éducatives. Cette tâche complexe implique la prise en compte de multiples contraintes, telles que la disponibilité des salles, les horaires des enseignants, les matières à couvrir, ainsi que les préférences et les disponibilités des enseignants et étudiants. Une mauvaise gestion peut entraîner des conflits d'horaires, une mauvaise utilisation des ressources et une insatisfaction générale parmi les parties prenantes. En particulier, l'Algorithme Intelligent Water Drops (IWD), pour optimiser la création des emplois du temps des examens.

L'objectif de ce mémoire est de concevoir un système intelligent pour générer des emplois du temps optimisés, minimisant les conflits et maximisant l'utilisation efficace des ressources. L'Algorithme Intelligent Water Drops, inspiré par le comportement naturel des rivières d'eau qui trouvent des chemins optimaux, sera adapté pour résoudre le problème de planification des examens. Cette méthode vise à fournir une solution robuste et flexible, capable de s'adapter aux besoins évolutifs des institutions académiques.

Organisation du mémoire

- **Chapitre 1 : Les Méthodes Intelligentes des Emplois du Temps**

Ce chapitre explore les différentes méthodes intelligentes et les techniques actuelles utilisées pour la gestion et l'optimisation des emplois du temps académique. Il couvre les approches traditionnelles ainsi que les innovations récentes dans le domaine, en mettant l'accent sur les avantages et les inconvénients de chaque méthode. La revue de la littérature permettra de situer l'Algorithme Intelligent Water Drops dans le contexte des autres techniques d'optimisation et de justifier son choix pour cette étude.

- **Chapitre 2 : Algorithme IWD pour le problème de calendrier d'examen**

Ce chapitre explore les emplois du temps académiques et introduit l'Algorithme Intelligent Water Drops (IWD) ainsi que ses variantes hybrides pour optimiser la planification des examens universitaires.

- **Chapitre 3 : Expérimentations et Résultats**

Le dernier chapitre présente les expérimentations réalisées pour évaluer la performance du système proposé. La base de données Les résultats des tests sur des données réelles et

synthétiques seront analysés pour déterminer l'efficacité de l'algorithme en termes de réduction des conflits d'horaires et d'optimisation de l'utilisation des ressources.

Chapitre I

Méthodes Pour La Gestion

Intelligente Des Emplois Du Temps

1. Introduction

La gestion intelligente des emplois du temps se développe grâce à une gamme de méthodes avancées qui exploitent les capacités de l'intelligence artificielle et des techniques d'optimisation. Ces méthodes incluent les algorithmes génétiques, qui simulent l'évolution naturelle pour trouver des solutions efficaces, les algorithmes de colonies de fourmis, qui imitent le comportement de recherche de nourriture des fourmis pour résoudre des problèmes de routage, et les algorithmes de recherche locale, qui explorent de manière itérative l'espace des solutions pour atteindre un optimum local. Chacune de ces approches est adaptée pour résoudre des problèmes spécifiques de planification des activités, en tenant compte des contraintes de ressources, des préférences des utilisateurs et des exigences spécifiques du contexte d'application. Ces méthodes représentent une avancée significative dans la gestion moderne des horaires, visant à optimiser l'utilisation des ressources tout en minimisant les conflits et en répondant efficacement aux besoins complexes des organisations.

2. Les algorithmes d'ordonnement

Le problème d'ordonnement est un défi de prise de décision essentiel dans divers domaines, allant de la fabrication à la logistique en passant par l'éducation. Il consiste à planifier efficacement les tâches et les ressources pour maximiser la productivité et minimiser les coûts. Par exemple, dans un environnement de fabrication[1], il s'agit de générer un plan détaillé des opérations à effectuer sur différentes machines tout en respectant les contraintes de temps et de ressources.

Ce processus est crucial pour optimiser la performance opérationnelle et peut être associé à divers objectifs tels que la minimisation du temps total de traitement, la réduction des retards ou la maximisation de l'utilisation des ressources.

La planification peut être statique, où toutes les tâches sont connues à l'avance, ou dynamique, où de nouvelles tâches peuvent apparaître ou être annulées en cours de route. Dans ce contexte, les problèmes sont classés en différentes catégories, telles que le flow shop, le job shop et l'atelier ouvert, en fonction de la nature des tâches et des contraintes d'ordonnement.

La résolution efficace de ces problèmes nécessite souvent l'utilisation d'algorithmes avancés, de modèles mathématiques et de systèmes d'aide à la décision pour optimiser les plans dans des environnements complexes et dynamiques [2]. En particulier, le job shop, qui implique des tâches avec des opérations spécifiques et des contraintes d'ordonnement complexes, est l'une des catégories les plus étudiées et utilisées dans la pratique.

Il existe de nombreux algorithmes et approches pour résoudre les problèmes d'ordonnancement, chacun adapté à des situations spécifiques et répondant à des objectifs particuliers. Voici quelques-uns des principaux types d'algorithmes d'ordonnancement :

. Ordonnancement basé sur les règles

L'ordonnancement basé sur les règles est une méthode classique de gestion des emplois du temps qui repose sur l'application de règles prédéfinies pour planifier les activités ou les tâches. Ces règles sont généralement conçues pour optimiser certains critères spécifiques, tels que la réduction des temps d'attente, la minimisation des conflits d'horaires ou la maximisation de l'utilisation des ressources [3]. Voici quelques exemples de règles couramment utilisées dans l'ordonnancement basé sur les règles :

1. Priorité des tâches : Les tâches ou les activités sont ordonnées en fonction de leur priorité. Les tâches les plus importantes ou les plus urgentes sont planifiées en premier, tandis que les tâches moins prioritaires sont planifiées ultérieurement [3]. Cette règle vise à garantir que les activités les plus critiques sont traitées en premier.

2. Disponibilité des ressources : Les activités sont planifiées en fonction de la disponibilité des ressources requises pour les exécuter. Cette règle garantit que les ressources nécessaires sont disponibles au moment où les activités doivent être effectuées, minimisant ainsi les retards et les interruptions.

3. Regroupement des tâches similaires : Les tâches similaires sont regroupées ou séquentiellement organisées pour optimiser l'utilisation des ressources et réduire les temps morts entre les activités. Par exemple, les rendez-vous avec des clients peuvent être regroupés par emplacement géographique pour minimiser les déplacements.

4. Minimisation des temps de traitement : Les activités sont ordonnées de manière à minimiser les temps de traitement total ou les temps d'attente. Cette règle vise à optimiser l'efficacité globale du processus en réduisant les temps morts et en maximisant le débit des tâches.

5. Préférences des utilisateurs : Les activités sont planifiées en tenant compte des préférences individuelles des utilisateurs ou des contraintes spécifiques, telles que les horaires de travail préférés ou les jours de congé [3]. Cette règle vise à maximiser la satisfaction des utilisateurs en leur offrant des horaires qui correspondent à leurs besoins et à leurs préférences.

L'ordonnancement basé sur les règles est souvent utilisé dans des environnements où les problèmes d'ordonnancement sont relativement simples et où les contraintes sont bien définies.

Cependant, cette approche peut être limitée par sa rigidité et son manque d'adaptabilité aux changements imprévus ou aux environnements dynamiques. C'est pourquoi il est souvent combiné avec d'autres méthodes d'ordonnancement plus sophistiquées, telles que les approches basées sur l'intelligence artificielle, pour obtenir de meilleurs résultats dans des situations plus complexes.

Algorithme de Johnson

L'algorithme de Johnson est un algorithme classique utilisé dans le domaine de l'ordonnancement pour minimiser le temps total de production dans les ateliers de fabrication à deux machines. Il est particulièrement efficace pour les problèmes où chaque tâche doit être traitée successivement sur chacune des deux machines [4], sans possibilité de réordonnement.

Voici les étapes de l'algorithme de Johnson :

1. Identification des temps de traitement : Pour chaque tâche, les temps de traitement sur les deux machines sont identifiés.
2. Classification des tâches : Les tâches sont classées en deux groupes : celles qui ont un temps de traitement plus court sur la première machine et celles qui ont un temps de traitement plus court sur la deuxième machine.
3. Ordonnement des tâches : Les tâches sont ordonnées en fonction de leur temps de traitement le plus court, soit sur la première machine, soit sur la deuxième machine. Les tâches du premier groupe sont placées en premier dans l'ordre croissant de leurs temps de traitement sur la première machine, tandis que les tâches du deuxième groupe sont placées en dernier dans l'ordre décroissant de leurs temps de traitement sur la deuxième machine.
4. Séquence finale : Une fois que les tâches sont ordonnées de cette manière, la séquence finale est obtenue en alternant les tâches des deux groupes dans l'ordre établi.

L'algorithme de Johnson garantit une séquence optimale pour minimiser le temps total de production dans les ateliers à deux machines, sous l'hypothèse que les temps de traitement sur les machines sont indépendants les uns des autres et que chaque tâche doit être traitée sur chacune des deux machines [4].

Cependant, il convient de noter que l'algorithme de Johnson est spécifique aux problèmes à deux machines et ne peut pas être généralisé à des problèmes comportant un nombre quelconque de

machines. De plus, il ne prend pas en compte d'autres facteurs tels que les contraintes de capacité ou les priorités des tâches, qui peuvent être importants dans des contextes plus complexes.

Algorithme de séquence critique

L'algorithme de séquence critique, également connu sous le nom de méthode du chemin critique, est une technique utilisée dans la gestion de projet pour déterminer les tâches qui ont le plus grand impact sur la durée totale du projet. L'objectif est d'identifier les tâches qui ne peuvent pas être retardées sans retarder le projet dans son ensemble [5]. Voici les étapes de l'algorithme de séquence critique :

1. Création du réseau de tâches : Les différentes tâches du projet sont représentées sous forme de nœuds dans un réseau, tandis que les dépendances entre les tâches sont représentées par des arcs [5].

2. Calcul des dates au plus tôt (DAT) et au plus tard (DPT) : Pour chaque tâche, on calcule deux dates : la date au plus tôt (DAT), qui représente le moment où la tâche peut commencer sans retarder le projet, et la date au plus tard (DPT), qui représente le moment où la tâche doit commencer pour ne pas retarder le projet. Ces dates sont calculées en utilisant les durées des tâches et les dépendances entre elles [5].

3. Détermination du chemin critique : Le chemin critique est le chemin le plus long à travers le réseau de tâches. Il est déterminé en identifiant les chemins qui ont la même durée totale que le chemin le plus long. Les tâches qui appartiennent à ce chemin critique sont celles qui ont un impact direct sur la durée totale du projet [5].

4. Analyse des marges de flottement : Les marges de flottement sont calculées pour chaque tâche non critique, ce qui représente la marge de temps dont une tâche peut bénéficier sans retarder le projet dans son ensemble. Cela permet de déterminer quelles tâches peuvent être retardées sans affecter la date de fin du projet.

5. Gestion du projet : Une fois le chemin critique identifié, la gestion du projet se concentre sur la gestion des tâches qui font partie de ce chemin, car ce sont celles qui ont le plus grand impact sur la durée totale du projet. Les autres tâches peuvent être gérées avec plus de flexibilité en fonction des marges de flottement disponibles.

L'algorithme de séquence critique est largement utilisé dans la planification et la gestion de projet pour identifier les tâches critiques et allouer efficacement les ressources pour garantir le

respect des délais. Il permet aux gestionnaires de projet de se concentrer sur les tâches les plus importantes et de prendre des décisions éclairées pour optimiser la durée du projet.

Programmation linéaire/entière

La programmation linéaire (PL) et la programmation linéaire entière (PLE) sont des techniques d'optimisation utilisées dans la gestion des emplois du temps et dans de nombreux autres domaines pour résoudre des problèmes d'optimisation avec des contraintes linéaires [6]. Voici une explication de ces deux approches :

1. Programmation Linéaire (PL) :

La programmation linéaire vise à maximiser ou minimiser une fonction linéaire, appelée fonction objectif, sous contraintes linéaires [7]. La fonction objective et les contraintes sont généralement exprimées sous forme de fonctions linéaires des variables de décision.

La forme générale d'un problème de programmation linéaire est la suivante :

Minimiser (ou maximiser) $Z = C_1X_1 + C_2X_2 + \dots + C_nX_n$

Les solutions optimales des problèmes de programmation linéaire sont généralement des valeurs continues des variables de décision.

2. Programmation Linéaire Entière (PLE) :

La programmation linéaire entière est une extension de la programmation linéaire dans laquelle les variables de décision doivent prendre des valeurs entières. Cette contrainte supplémentaire rend souvent les problèmes beaucoup plus difficiles à résoudre [8], car l'espace des solutions devient discret et discontinu.

Les problèmes de programmation linéaire entière peuvent être formulés de la même manière que les problèmes de programmation linéaire, mais avec l'ajout de contraintes de non-négativité et de contraintes de prise de valeurs entières pour certaines ou toutes les variables de décision [8].

Par exemple, un problème de PLE peut être formulé comme suit :

Minimiser (ou maximiser) $Z = C_1X_1 + C_2X_2 + \dots + C_nX_n$

La programmation linéaire et la programmation linéaire entière sont utilisées dans de nombreux contextes de gestion des emplois du temps pour résoudre des problèmes tels que l'affectation de ressources, la planification des horaires de travail, l'optimisation des itinéraires, etc [8]. Elles offrent des méthodes puissantes pour trouver des solutions efficaces tout en respectant les contraintes spécifiques du problème.

Ces algorithmes peuvent être adaptés en fonction des caractéristiques spécifiques du problème d'ordonnement, telles que le nombre de tâches, les contraintes de ressources et les objectifs d'optimisation. En pratique, le choix de l'algorithme dépend souvent de la complexité du problème et des performances requises en termes de temps de calcul et de qualité de la solution.

3. L'apprentissage automatique [TomMichel]

L'apprentissage automatique est la science qui consiste à programmer des ordinateurs pour qu'ils apprennent à partir de données. Arthur Samuel a été le premier à définir l'apprentissage automatique comme le domaine scientifique qui donne aux ordinateurs la capacité d'apprendre sans être programmés explicitement, c'est-à-dire qu'au lieu d'exécuter des instructions préprogrammées comme n'importe quel algorithme générique, un algorithme d'apprentissage automatique est capable d'apprendre à partir de l'expérience et, en particulier, des données qui lui sont fournies. Tom Michel a proposé une définition plus orientée vers l'ingénierie, établissant qu'"un programme peut être classé comme apprenant à partir de l'expérience E, liée à une classe de tâches T avec un indicateur de performance D, si sa performance sur la tâche T, mesurée par D, s'améliore avec l'expérience E". De manière pragmatique, l'apprentissage automatique est un sous-domaine de l'informatique qui développe des algorithmes dépendant des données. Les données peuvent avoir les origines les plus diverses, allant d'autres systèmes informatiques à la nature elle-même ou même être générées par l'homme. En tant que telles, elles peuvent être utilisées dans presque tous les domaines (par exemple, en médecine, elles peuvent être appliquées pour analyser de grandes quantités d'images médicales afin d'établir un diagnostic). En résumé, l'apprentissage automatique est le processus

de résolution d'un problème par la collecte d'un ensemble de données (communément appelé "jeu de données") et, par l'utilisation d'algorithmes spécifiques, la création d'un modèle statistique de ces données.

- Le domaine de l'apprentissage automatique peut être divisé en trois catégories : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

L'apprentissage supervisé

Dans le cas de l'apprentissage supervisé, l'ensemble de données est composé uniquement d'exemples étiquetés, selon la formule suivante : $(x_i, y_i)_{i=1}^N$ où chaque élément x_i dans N est un vecteur d'attributs. Un vecteur d'attributs est un vecteur dont chaque dimension $j=1, \dots$, contient une valeur qui décrit l'exemple d'une manière ou d'une autre. Cette valeur est un attribut (également appelé caractéristique) et est indiquée par (j) . Pour tous les exemples de l'ensemble de données, il est garanti que l'attribut situé à la position j du vecteur d'attributs contient toujours le même type d'informations. L'étiquette y_i est généralement un élément qui appartient à un certain ensemble de classes $\{1, 2, \dots, C\}$ soit un nombre réel [9]. Certaines approches utilisent des structures plus complexes pour les étiquettes, telles que des vecteurs, des matrices ou même des graphes, mais elles sont assez rares et très spécifiques à certains problèmes. L'objectif d'un algorithme d'apprentissage supervisé est d'utiliser l'ensemble de données pour produire un modèle qui reçoit le vecteur d'attributs x en entrée et qui donne, en sortie, des informations permettant de déduire l'étiquette de ce vecteur.

L'apprentissage non supervisé

L'apprentissage non supervisé diffère de l'apprentissage supervisé en ce sens qu'il utilise un ensemble de données dont les exemples ne sont pas étiquetés, $(x_i)_{i=1}^N$ mais x reste un vecteur d'attributs. L'objectif de l'apprentissage non supervisé est la création d'un modèle qui reçoit en entrée le vecteur d'attributs et le transforme en un autre vecteur ou en une valeur qui peut être utile d'une autre manière pour résoudre un problème [8]. Par exemple, les techniques de regroupement créent des modèles qui calculent l'identifiant de regroupement pour chaque vecteur d'attributs dans l'ensemble de données ; les techniques de détection des anomalies (ou détection des valeurs aberrantes) calculent une valeur qui détermine à quel point un exemple x est différent d'un élément typique de l'ensemble de données ; et les techniques de réduction de la dimensionnalité produisent un vecteur d'attributs qui a moins d'attributs que le vecteur original, cette réduction du nombre d'attributs le rendant plus facile à interpréter.

L'apprentissage par renforcement

L'apprentissage par renforcement est très différent des autres catégories. D'un point de vue pragmatique, on pourrait s'attendre à ce que l'apprentissage supervisé et non supervisé couvre toutes les applications possibles de ces techniques : lorsqu'il y a des données étiquetées, on utilise l'apprentissage supervisé, lorsqu'il n'y a pas de données étiquetées, on utilise l'apprentissage non supervisé, et lorsque les deux situations sont vérifiées, on applique des techniques semi-supervisées. Cependant, ce n'est pas tout à fait vrai. Bien que l'apprentissage par renforcement semble similaire à l'apprentissage non supervisé puisqu'il n'utilise pas d'exemples étiquetés présentant le comportement correct, il vise à maximiser une récompense et non à rechercher une structure cachée. La découverte d'une telle structure peut même être une expérience utile dans sa résolution, mais elle ne résout pas à elle seule le problème de la récompense. Par conséquent, l'apprentissage par renforcement est considéré comme une catégorie différente de problèmes d'apprentissage automatique [9]. Compte tenu de son importance pour ce travail, cette catégorie est détaillée dans la section suivante.

- Il est possible de trouver dans la littérature scientifique des références à une quatrième catégorie : **l'apprentissage semi-supervisé**. Il se situe entre l'apprentissage supervisé et l'apprentissage non supervisé et se caractérise par un ensemble de données contenant à la fois des exemples étiquetés et non étiquetés. En général, le nombre d'exemples étiquetés est beaucoup plus important que le nombre d'exemples non étiquetés. L'objectif d'un algorithme d'apprentissage semi-supervisé est similaire à celui d'un algorithme d'apprentissage supervisé, avec l'espoir que l'utilisation de données non étiquetées puisse aider l'algorithme à produire un meilleur modèle. Il n'est pas évident que l'apprentissage supervisé puisse bénéficier de l'ajout de données non étiquetées, car cela augmente l'incertitude du problème [9]. Cependant, lorsque ces exemples sont ajoutés, les informations disponibles sur le problème augmentent, ce qui améliore la distribution de probabilité des données. Un algorithme d'apprentissage semi-supervisé doit pouvoir utiliser ces informations supplémentaires pour améliorer la précision de son modèle. Parmi les exemples, citons l'auto-apprentissage, le co-apprentissage et le tri-apprentissage.

4. Systems experts

Les systèmes experts sont une forme d'intelligence artificielle qui simulent le raisonnement et l'expertise d'un expert humain dans un domaine spécifique. Ils se composent d'une base de connaissances et d'un moteur d'inférence qui applique un raisonnement logique pour résoudre des problèmes. Contrairement aux programmes conventionnels, les systèmes

experts peuvent fonctionner avec des informations incomplètes et faire des erreurs, comme un expert humain [10]. Ils sont conçus pour être interactifs et fiables dans la prise de décision.

Les principaux avantages des systèmes experts sont:

- Amélioration de la qualité des décisions
- Réduction des coûts liés à la consultation d'experts
- Rapidité et cohérence des solutions
- Fiabilité

Cependant, les systèmes experts ont aussi des limites :

- Ils ne peuvent pas fournir de solutions créatives comme un expert humain
- Ils peuvent être coûteux à maintenir et à mettre à jour

Les systèmes experts sont utilisés dans de nombreux domaines comme la finance, la médecine, la gestion des ressources humaines, etc. Leur développement nécessite la collaboration entre un expert du domaine et un ingénieur des connaissances. En résumé, les systèmes experts sont une forme d'IA qui simulent l'expertise humaine pour résoudre des problèmes de manière fiable et rapide, bien qu'ils aient des limites par rapport à la créativité d'un expert humain.

Comment fonctionne un système expert pour la gestion des emplois du temps

un système expert pour la gestion optimisée des emplois du temps fonctionne de la manière suivante :

1. Le système expert combine une base de connaissances, qui contient l'expertise des experts en gestion des emplois du temps, et un moteur d'inférence, qui utilise des algorithmes d'optimisation pour générer des solutions optimales¹.
2. La base de connaissances contient des informations sur les contraintes, les préférences et les règles à respecter pour la création des emplois du temps, comme les disponibilités des professeurs, les créneaux horaires, les salles de cours, etc [11].
3. Le moteur d'inférence du système expert utilise des techniques avancées comme le chaînage avant, le chaînage arrière ou des algorithmes génétiques pour explorer l'espace des solutions possibles et trouver l'ordonnancement optimal des tâches et des ressources.
4. Le système expert peut être couplé à une base de données pour une gestion efficace des données liées aux emplois du temps (professeurs, matières, créneaux horaires, salles, etc.).

5. Certains systèmes experts sont conçus pour être multi-experts, avec plusieurs modules spécialisés qui coopèrent pour résoudre le problème de manière plus complète. Le contrôle peut être centralisé ou décentralisé.

6. Les systèmes experts permettent de faire évoluer les connaissances au fur et à mesure de leur utilisation, en intégrant les retours d'expérience des utilisateurs.

En résumé, le système expert pour la gestion des emplois du temps combine une base de connaissances sur les contraintes et règles du domaine avec un moteur d'inférence utilisant des algorithmes d'optimisation avancés, le tout étant souvent couplé à une base de données pour une gestion efficace des informations.

5. Optimisation multi-objectif

Le problème de gestion des emplois du temps est considéré comme un problème d'optimisation combinatoire de nature multi-objectif et de la classe NP-complet. Cela signifie qu'il comporte plusieurs objectifs potentiellement contradictoires à optimiser simultanément, et qu'il est difficile à résoudre de manière optimale [12].

Les principaux objectifs à optimiser dans la gestion des emplois du temps

- Maximiser le nombre de demi-journées libres pour les professeurs/classes
- Minimiser le nombre de trous (périodes libres) dans les emplois du temps
- Éviter les cours isolés (cours seuls dans une demi-journée)

Les différentes approches d'optimisation multi objectifs

Pour résoudre ce problème multi-objectif, différentes approches d'optimisation multi-objectif ont été étudiées :

- Les méthodes par transformation en problème mono-objectif
- Les méthodes non Pareto
- Les méthodes Pareto, qui cherchent à trouver un ensemble de solutions Pareto-optimales

Les méthodes Pareto semblent être les plus utilisées, car elles permettent de produire un ensemble de solutions de compromis sans avoir à fixer a priori les priorités entre les différents objectifs. Des algorithmes comme NSGA-II ont été proposés pour résoudre le problème d'emploi du temps de manière multi-objectif [12]. Ils maintiennent une population de solutions Pareto-optimales au cours de la recherche. L'optimisation multi-objectif permet d'améliorer significativement la qualité des emplois du temps en termes de nombre de demi-journées libres, de trous et de cours isolés. Elle ne nécessite pas d'effort particulier, il suffit de laisser l'ordinateur lancer l'optimisation qui s'effectue en temps réel. En résumé, l'optimisation multi-

objectif est une approche pertinente pour résoudre le problème complexe de gestion des emplois du temps, en permettant de trouver un ensemble de solutions de compromis entre les différents objectifs à optimiser.

6. Agents intelligents

Les agents intelligents pour la gestion des emplois du temps sont des entités logicielles qui peuvent prendre des décisions et agir de manière autonome dans un environnement complexe. Ils sont conçus pour résoudre des problèmes de gestion des emplois du temps, tels que la planification des cours, la gestion des ressources, la résolution de conflits, etc [13].

Les catégories de l'agents intelligents

Les agents intelligents pour la gestion des emplois du temps peuvent être divisés en plusieurs catégories, notamment :

Agents réactifs

Ils sont conçus pour réagir rapidement à des problèmes simples et ne nécessitent pas un haut niveau de raisonnement. Ils sont souvent utilisés dans des systèmes de gestion des emplois du temps qui nécessitent une rapidité de réponse [12].

Agents cognitifs

Ils sont fondés sur la coopération d'agents capables, à eux seuls, d'effectuer des opérations complexes. Chaque agent dispose d'une capacité de raisonnement, d'une aptitude à traiter des informations diverses liées au domaine d'application, et d'informations relatives à la gestion des interactions avec d'autres agents et l'environnement [12].

Agents hybrides

Ils sont composés d'un ensemble de modules organisés dans une hiérarchie, chaque module étant soit une composante cognitive avec représentation symbolique des connaissances et capacités de raisonnement, soit une composante réactive [13].

Les agents intelligents pour la gestion des emplois du temps peuvent être utilisés dans différents contextes, tels que :

- **Gestion des emplois du temps dans les universités** : Les agents intelligents peuvent être utilisés pour résoudre le problème d'emploi du temps dans les universités, en prenant en compte les contraintes telles que les disponibilités des professeurs, les créneaux horaires, les salles de cours, etc.
- **Gestion des emplois du temps dans les entreprises** : Les agents intelligents peuvent être utilisés pour résoudre le problème d'emploi du temps dans les entreprises, en prenant en compte

les contraintes telles que les disponibilités des employés, les créneaux horaires, les ressources, etc.

- **Gestion des emplois du temps dans les systèmes de santé** : Les agents intelligents peuvent être utilisés pour résoudre le problème d'emploi du temps dans les systèmes de santé, en prenant en compte les contraintes telles que les disponibilités des médecins, les créneaux horaires, les salles de consultation, etc.

Les avantages des agents intelligents pour la gestion des emplois du temps

- **Meilleure efficacité** : Les agents intelligents peuvent résoudre des problèmes de gestion des emplois du temps de manière plus efficace que les méthodes traditionnelles.
- **Meilleure flexibilité** : Les agents intelligents peuvent être adaptés à différents contextes et peuvent prendre en compte des contraintes spécifiques.
- **Meilleure rapidité** : Les agents intelligents peuvent résoudre des problèmes de gestion des emplois du temps de manière plus rapide que les méthodes traditionnelles.

Les inconvénients des agents intelligents pour la gestion des emplois du temps

- **Complexité** : Les agents intelligents peuvent être complexes à mettre en œuvre et à paramétrer.
- **Coûts** : Les agents intelligents peuvent nécessiter des ressources importantes, notamment en termes de développement et de maintenance.
- **Dépendance à l'implémentation** : Les agents intelligents peuvent dépendre de l'implémentation spécifique et peuvent ne pas être généralisables à d'autres contextes.

En résumé, les agents intelligents pour la gestion des emplois du temps sont des outils puissants qui peuvent résoudre des problèmes complexes de gestion des emplois du temps de manière efficace, flexible et rapide. Cependant, ils peuvent également présenter des inconvénients tels que la complexité, les coûts et la dépendance à l'implémentation. Des modèles comme MATP (Multi-Agent model for Timetabling Problem) ont été proposés pour résoudre le problème d'emploi du temps universitaire de manière décentralisée et parallèle. Ils permettent d'intégrer de nouveaux détails et contraintes qui n'avaient pas été pris en compte auparavant. Les résultats expérimentaux montrent que ces approches multi-agents sont plus efficaces en termes de temps d'exécution et de qualité des solutions générées, notamment grâce à la prise en compte de contraintes "soft" qui améliorent la qualité des emplois du temps. En résumé, les systèmes multi-agents avec des agents intelligents représentent une solution intéressante pour résoudre le

problème complexe de gestion des emplois du temps, en permettant un traitement décentralisé et une meilleure prise en compte des contraintes.

7. Les méthodes heuristiques [Troudi,2006]

Les méthodes heuristiques sont des algorithmes qui permettent de trouver rapidement une solution réalisable, mais pas nécessairement optimale, pour des problèmes d'optimisation complexes comme la gestion des emplois du temps. Elles s'appuient sur la connaissance du problème pour définir une stratégie de résolution. Parmi les principales méthodes heuristiques utilisées pour la gestion des emplois du temps, on peut citer :

Les algorithmes gloutons

Un algorithme glouton est une méthode d'optimisation qui consiste à faire des choix localement optimaux à chaque étape, dans l'espoir d'aboutir à une solution globalement optimale. Contrairement à d'autres approches plus complexes, les algorithmes gloutons prennent des décisions immédiates en se basant sur des critères locaux sans considérer les conséquences à long terme. Dans le contexte de la gestion des emplois du temps [14], un algorithme glouton peut être utilisé pour résoudre des problèmes tels que l'organisation des activités dans une salle unique. Par exemple, pour maximiser le nombre d'activités organisées dans une salle sans qu'elles se chevauchent, un algorithme glouton pourrait sélectionner les activités en fonction de critères locaux tels que l'heure de début, la durée ou le nombre de conflits potentiels avec d'autres activités.

L'algorithme glouton pour la gestion des emplois du temps peut être utilisé pour des tâches telles que la planification des cours, la gestion des ressources ou l'organisation des événements, en prenant des décisions itératives basées sur des critères locaux pour atteindre un objectif global, comme maximiser l'utilisation de la salle ou minimiser les conflits d'emploi du temps.

Les méthodes de recherche locale [Troudi,2006]

La recherche locale est une méthode d'optimisation qui consiste à explorer l'espace des solutions de manière séquentielle, en passant d'une étape de la solution courante à une solution "proche". Cette approche est utilisée pour résoudre des problèmes d'optimisation combinatoire sans contraintes, en cherchant à améliorer la solution courante en remplaçant la solution actuelle par une solution voisine si elle est meilleure. Dans le contexte de la recherche d'emploi, la recherche locale peut être utilisée pour trouver des emplois à proximité d'un emplacement géographique spécifique. Les moteurs de recherche, tels que Google, utilisent des algorithmes de recherche locale pour proposer des résultats adaptés à la localisation géographique de l'utilisateur. Cela signifie que les résultats de recherche sont influencés par la localisation

géographique de l'utilisateur, ce qui permet aux utilisateurs de trouver des emplois à proximité de leur emplacement. La recherche locale est également utilisée dans le contexte de la recherche d'immobilier, où elle consiste à trouver un local commercial ou résidentiel à proximité d'un emplacement géographique spécifique. Les agents immobiliers et les sites internet spécialisés utilisent des méthodologies spécifiques pour trouver des locaux qui correspondent aux critères de recherche des clients. En résumé, la recherche locale est une méthode d'optimisation qui consiste à explorer l'espace des solutions de manière séquentielle, en passant d'une étape de la solution courante à une solution "proche". Elle est utilisée dans de nombreux domaines, notamment dans la recherche d'emploi et d'immobilier, pour trouver des solutions adaptées à la localisation géographique de l'utilisateur.

Les algorithmes génétiques

Un algorithme génétique est une méthode d'optimisation inspirée par le processus d'évolution naturelle des espèces. Il fonctionne en créant une population initiale de solutions candidates, évaluant ces solutions en fonction d'une fonction d'aptitude, sélectionnant les meilleures solutions pour se reproduire et créer de nouveaux individus, et répétant ce processus itérativement pour améliorer la population. Dans le contexte de la gestion des emplois du temps, un algorithme génétique peut être utilisé pour optimiser la planification des horaires en prenant en compte divers critères tels que les disponibilités des employés, les contraintes de temps, les préférences individuelles, etc [15]. Il fonctionne en générant une population initiale de solutions potentielles pour les emplois du temps, évaluant ces solutions en fonction de leur adéquation aux critères définis, et en évoluant progressivement vers des solutions de meilleure qualité à travers des opérations de sélection, de croisement et de mutation. Ce processus itératif permet d'obtenir des emplois du temps optimisés en tenant compte de multiples contraintes et objectifs.

Les méthodes hybrides

La gestion hybride des emplois du temps est une approche qui combine différents modes de travail, tels que le travail en présentiel et le télétravail, pour offrir une flexibilité aux employés tout en assurant une organisation efficace du temps de travail. Cette méthode permet aux entreprises de tirer parti des avantages du travail à distance tout en maintenant une certaine présence au bureau [16]. Fonctionnement de la gestion hybride des emplois du temps :

1. Flexibilité : Les employés ont la possibilité de travailler à distance ou au bureau en fonction de leurs besoins et des exigences de leur poste.

Chapitre I Méthodes Pour La Gestion Intelligente Des Emplois Du Temps

2. Planification adaptative : Les équipes peuvent organiser leur emploi du temps en fonction des tâches à réaliser, en alliant des moments de travail collaboratif en présentiel et des périodes de concentration individuelle en télétravail.

3. Communication et collaboration : Les outils de communication et de collaboration en ligne sont essentiels pour maintenir la cohésion d'équipe et faciliter le travail à distance.

4. Suivi du temps de travail : Il est important d'avoir des outils de suivi du temps de travail adaptés pour garantir le respect des horaires et la mesure de la productivité, que ce soit en présentiel ou en télétravail.

5. Adaptabilité : La gestion hybride des emplois du temps nécessite une certaine adaptabilité de la part des employés et des managers pour s'assurer que les objectifs sont atteints tout en respectant les besoins individuels.

En résumé, la gestion hybride des emplois du temps combine les avantages du travail en présentiel et du télétravail pour offrir flexibilité, efficacité et équilibre entre vie professionnelle et personnelle. Elle repose sur une organisation adaptative, une communication efficace et une utilisation judicieuse des outils de suivi du temps de travail.

Ces méthodes heuristiques sont généralement plus rapides que les méthodes exactes, mais ne garantissent pas l'optimalité de la solution. Elles sont donc particulièrement adaptées aux problèmes de gestion des emplois du temps, qui sont de nature combinatoire et NP-complets²⁴. L'utilisation de ces méthodes heuristiques, souvent implémentées dans des logiciels dédiés, permet d'automatiser la génération d'emplois du temps et d'obtenir des solutions de bonne qualité en un temps.

8. Conclusion

En conclusion, les méthodes intelligentes de gestion des emplois du temps apportent une solution innovante et efficace aux défis complexes de la planification des activités dans divers contextes. En utilisant des techniques avancées d'intelligence artificielle telles que les algorithmes génétiques, les algorithmes de colonies de fourmis et les algorithmes de recherche locale, ces approches permettent de créer des emplois du temps optimisés qui minimisent les conflits et maximisent l'utilisation des ressources disponibles. Grâce à leur capacité à prendre en compte de multiples contraintes et préférences, ces méthodes offrent une flexibilité et une adaptabilité cruciales pour répondre aux besoins dynamiques des institutions académiques et des organisations professionnelles. En intégrant ces technologies, la gestion des emplois du temps devient non seulement plus efficace, mais aussi plus réactive et personnalisée, ouvrant la voie à des améliorations continues et à des gains de productivité significatifs.

Chapitre II

*Algorithme IWD pour le problème de
calendrier d'examen*

1 Introduction

L'objectif de cette étude est d'analyser l'algorithme métaheuristique Intelligent Water Drops (IWD) afin de créer et de générer des solutions de qualité supérieure pour résoudre le problème de planification des examens universitaires. La métaheuristique basée sur la population, connue sous le nom d'IWD, simule la dynamique des réseaux fluviaux. L'étude de l'algorithme IWD pour le problème de calendrier des examens est principalement motivée par la capacité à explorer de manière efficace l'espace de recherche. L'inconvénient majeur de l'algorithme IWD réside, tout comme les autres algorithmes basés sur la population, dans son exploitation, où il peut analyser de manière très efficace plusieurs niches d'espace de recherche, mais il ne peut pas explorer chaque niche vers laquelle il se dirige. Ce chapitre présente une méthode hybride qui combine l'algorithme de recherche IWD et locale afin d'améliorer l'exploitation.

2 Généralités sur les emplois du temps

Définition d'emplois du temps [Matthieu, 2008]

Les emplois du temps sont des calendriers de travail, où figurent à la fois le temps, l'affectation du personnel, les jours et les horaires de travail, et les congés et repos. Les emplois du temps peuvent être utilisés pour planifier les horaires de présences du personnel ou les tâches effectuées par le personnel.

2.2 Historique

Les premières activités d'emploi du temps ont été effectuées manuellement et un emploi du temps typique, une fois construit est resté statique avec seulement quelques changements nécessaires. Cependant la nature des enseignements a changé considérablement au cours des années et ainsi les exigences en matière de confection d'emploi du temps sont devenues beaucoup plus compliquées qu'ils ont eu l'habitude de l'être. L'intérêt de génération d'emploi dutemps a augmenté dramatiquement dans les années 60 principalement en la raison de la disponibilité d'ordinateurs pour exécuter les algorithmes développés. Autour de la fin des années 60 quelques tentatives qui ont traité le problème en considérant des études de cas commençaient à être publiées. Par exemple en 1969, Lawrie a développé un modèle pour le problème de confection d'horaire en employant l'approche de programmation linéaire. Pendant les années 1970, plusieurs publications ont abordé le problème d'emploi du temps. Les principales techniques qui semblent avoir été plus répandu dans les années 1970 et les années 1980 sont les techniques ayant pour racine l'intelligence artificielle et sont basées sur les méthodes du recuit simulé, la recherche Tabou et les algorithmes génétiques. En 1985, De Werra, a décrit les divers

problèmes traitant le problème d'emploi du temps d'une façon formelle et a fourni les différentes formulations dans une tentative de les résoudre. Il a aussi décrit les approches considérées les plus importantes à ce temps-là. En 1986, Carter, a fait une analyse sur de réelles applications de confection d'emploi du temps de plusieurs universités. Junginger, a décrit dans la même année, les recherches faites en Allemagne sur le problème d'emploi du temps scolaires et les approches qui étaient basées sur des heuristiques directes, en particulier il a décrit les divers logiciels mis en œuvre et leur utilisation dans les divers établissements. En 1994, Corne, a fait une enquête sur l'application des algorithmes génétiques au problème d'emploi du temps et a controversé les futures perspectives de telles approches en comparant les résultats obtenus avec ceux obtenus avec d'autres approches. Dans les dernières décennies, les sujets de résolution du problème d'emploi du temps ont été principalement limités à la (RO) (les techniques employées étaient naturellement mathématiques). Dans la décennie actuelle, la contribution de l'IA a fourni au problème de résolution de l'emploi du temps une heuristique moderne telle que les algorithmes génétiques, le recuit simulé et la recherche Tabou.

A Quoi sert un emploi du temps [Matthieu,2008]

Depuis le début des années 80, la gestion des ressources humaines a été reconnue comme une activité stratégique pour un établissement. Avec cette reconnaissance, l'intérêt d'élaborer des emplois du temps s'est vu accroître de plus en plus car ils permettent :

- Aux établissements exerçant une activité continue ou quasi-continue de répartir convenablement leur personnel (universités, entreprises de transports, hôpitaux, etc...).
- Aux établissements cherchant à se rendre plus accessibles à la clientèle d'étaler les horaires d'ouverture (grands magasins, banques, etc...).
- A toutes les établissements de surmonter leur exigences de productivité et de mieux gérer les présences et absences de leur personnel.

Les situations où un emploi du temps est utile sont nombreuses. Elles justifient l'existence de différentes formes de ce dernier dans un même système : emplois du temps à court, moyen et à long terme.

Comment un emploi du temps est élaboré [Troudi ,2006]

Pour que les emplois du temps élaborés soient satisfaisants, ils doivent vérifier un ensemble de contraintes et établir un meilleur compromis entre les différents acteurs (exemple :le chef du département, le planificateur, les enseignants et les étudiants). Lorsque les différentes solutions alternatives sont connues, une négociation se déroule de la manière suivante : chaque acteur donne son opinion. Les points d'accord sont très vite expédiés et les points incertains sont débattus. Et des solutions de compromis sont dégagées. Les difficultés de négociation augmentent avec le nombre d'acteurs et le nombre de solutions alternatives. L'aspect combinatoire (pour l'élaboration des emplois du temps) rend d'autant plus difficile la négociation, car les opinions sont plus difficiles à formuler. Les moyens informatiques apportent une aide certaine notamment dans l'acquisition et la confrontation des données individuelles.

Les emplois du temps des examens universitaires

La planification des examens universitaires consiste à attribuer les examens à des créneaux horaires et à des salles en respectant des contraintes strictes, comme les conflits d'horaires, et des contraintes souples, comme la qualité du calendrier. La problématique de l'horaire d'examen peut être compliquée par la capacité des salles d'examen (Lei et al., 2018). Lorsque les capacités des salles sont prises en compte, il est impératif que le nombre d'étudiants assignés à une salle spécifique ne dépasse pas la capacité prévue. Toutefois, notre étude se concentre uniquement sur le problème de planification des examens universitaires sans tenir compte des capacités [17].

De nombreuses techniques d'approximation pour problème de planification des examens universitaires (PPEU) ont été développées par les communautés de l'intelligence artificielle (IA) et de la recherche opérationnelle (RO). Une enquête récente est disponible dans (Aldeeb et al., 2019). Une méthode courante consiste à utiliser l'heuristique de coloration des graphes, qui attribue les examens aux créneaux horaires un par un, en fonction de leur complexité. Pour ajuster le calendrier en cas d'examens non programmés, une méthode de retour en arrière est souvent utilisée en complément de ces techniques.

3. Utilisation de l'algorithme Intelligent Water Drops (IWD)

Définition de l'algorithme IWD

L'algorithme Intelligent Water Drops (IWD) est une optimisation inspirée de la nature basée sur les essais et introduite par Shah-Hosseini. IWD est un algorithme constructif qui construit une solution optimale grâce à la coopération entre un groupe d'agents appelés gouttes d'eau. L'algorithme imite le phénomène d'un essaim de gouttes d'eau coulant avec de la terre le long du lit d'une rivière. De manière procédurale, chaque goutte d'eau construit progressivement une solution à travers une série de transitions itératives d'un nœud au suivant jusqu'à ce qu'une solution complète soit obtenue [17]. Les gouttes d'eau communiquent entre elles via un attribut appelé sol, qui est associé au chemin entre deux points quelconques. La valeur du sol est utilisée pour déterminer la direction du mouvement du nœud actuel au suivant, un chemin avec une quantité de sol inférieure étant susceptible d'être suivi.

Application de l'algorithme IWD

L'algorithme IWD a été utilisé avec succès pour résoudre de nombreux problèmes d'optimisation combinatoire et continue provenant de différents domaines d'application. Il a été adopté pour résoudre des problèmes d'optimisation tels que l'optimisation des fonctions, le voyageur de commerce, les sacs à dos multiples, les problèmes de puzzle n-queen, la sélection de fonctionnalités, la planification de processeurs parallèles [17]. IWD a été utilisé avec succès pour résoudre des problèmes d'optimisation multi-objectifs. Certains efforts ont été déployés par les chercheurs pour étudier les aspects algorithmiques fondamentaux de l'IWD afin d'améliorer la capacité de recherche.

Performance de l'algorithme IWD [Shah-Hosseini, 2012]

L'algorithme Intelligent Water Drops (IWD) est une métaheuristique inspirée des comportements naturels des gouttes d'eau intelligentes lorsqu'elles coulent dans un environnement et cherchent un chemin optimal. Cet algorithme est souvent comparé à d'autres techniques d'optimisation en raison de sa capacité à résoudre efficacement divers problèmes complexes. Voici une analyse de sa performance et de ses caractéristiques clés :

1. Exploration et Exploitation

L'algorithme IWD est conçu pour équilibrer l'exploration de l'espace de recherche et l'exploitation des solutions prometteuses :

Chapitre II Algorithme IWD Pour Le Problème De Calendrier D'examen

- Exploration : Grâce à son mécanisme basé sur les comportements des gouttes d'eau, l'IWD explore largement l'espace de recherche, permettant de découvrir plusieurs zones potentielles contenant des solutions optimales.
- Exploitation : Les interactions locales entre les gouttes d'eau et l'environnement améliorent continuellement les solutions, favorisant la convergence vers des optima locaux.

2. Adaptabilité et Robustesse

L'IWD est adaptable à une variété de problèmes d'optimisation combinatoire, tels que :

- Problèmes de routage : Il a été utilisé avec succès pour résoudre des problèmes comme le problème de l'arbre de Steiner (STP) et le problème du voyageur de commerce (TSP).
- Planification et ordonnancement : Il peut optimiser des plannings d'examens, des ordonnancements de production et d'autres tâches nécessitant des solutions optimisées.

3. Efficacité en Convergence

- Convergence Rapide : L'algorithme IWD montre une vitesse de convergence rapide vers des solutions de haute qualité. Cela est dû à ses mécanismes d'itération où les gouttes d'eau s'adaptent et améliorent constamment leur chemin basé sur des retours itératifs.
- Optimisation Locale: En hybridant l'IWD avec des techniques de recherche locale, comme la programmation quadratique séquentielle ou l'algorithme de recherche locale itérée (ILS), la convergence vers l'optimum global est encore améliorée.

4. Comparaison avec d'Autres Algorithmes

- Performance Supérieure : Des études montrent que l'IWD hybride peut surpasser les heuristiques traditionnelles et d'autres algorithmes de recherche locale en termes de vitesse de convergence et de qualité des solutions.
- Résultats Expérimentaux Positifs : Les applications de l'IWD hybride dans divers domaines (comme la planification des commandes en production industrielle) ont démontré des améliorations significatives en termes de bénéfices et de réduction des délais.

5. Applications Réelles

- Problème de l'Arbre de Steiner (STP) : L'IWD hybride avec programmation quadratique séquentielle a montré des améliorations notables en termes de convergence et de performance par rapport aux versions non hybrides de l'IWD.

- Planification des Commandes : En combinant l'IWD avec l'ILS, l'algorithme a été capable de gérer efficacement des problèmes de planification des commandes avec des contraintes de capacité, améliorant les profits et minimisant les retards.

4. Hybrides algorithme

Définition de l'algorithme hybrides

L'algorithme IWD hybride (Intelligent Water Drops) combine la technique de l'IWD avec une méthode de recherche locale pour améliorer les performances globales de l'algorithme. L'hybridation vise à exploiter les avantages de l'algorithme IWD, qui est basé sur la population et capable d'explorer de larges régions de l'espace de recherche, avec les capacités de convergence rapide et d'optimisation locale des techniques de recherche locale. Par exemple, l'hybridation de l'IWD avec la programmation quadratique séquentielle ou l'algorithme de recherche locale itérée (ILS) permet d'optimiser les solutions locales tout en maintenant une exploration efficace de l'espace de recherche global [16].

Algorithme Hybride IWD proposé pour PPEU

L'algorithme IWD, malgré ses capacités de recherche globale et son taux de convergence satisfaisant, présente des limitations lorsqu'il est utilisé pour résoudre PPEU (AlDeeb, 2016). En effet, il tend à se piéger dans des optima locaux, ce qui limite son efficacité dans l'exploitation des zones de l'espace de recherche contenant les meilleures solutions. Pour surmonter cette faiblesse, un algorithme IWD hybride intégrant un optimiseur de recherche locale est proposé. Cet algorithme hybride, appelé "Hybrid IWD", vise à trouver un équilibre optimal entre exploration et exploitation.

Dans cette approche, les chercheurs décriront initialement les méthodes et les points d'intégration de l'optimiseur de recherche locale au sein de l'algorithme IWD. Ensuite, le processus d'optimisation par recherche locale sera détaillé, en insistant particulièrement sur la structure de voisinage la plus efficace utilisée.

IWD pour le problème de l'horaire d'examen sans capacité

L'algorithme IWD original a été adapté pour répondre aux spécificités et aux exigences de PPEU. La principale personnalisation consiste à maintenir la faisabilité pendant la recherche IWD en utilisant le degré de saturation, qui est intégré à l'algorithme IWD modifié [16].

La JIF adaptée comprend six phases principales, chacune comportant plusieurs étapes. Les phases et les étapes suivantes détaillent comment l'algorithme IWD modifié a été personnalisé pour être efficace pour PPEU.

Phase 1 : Initialisation des paramètres statiques

Cette phase implique la configuration initiale des paramètres statiques de l'algorithme IWD, c'est-à-dire ceux qui restent constants tout au long du processus de recherche. Voici une explication détaillée :

Les paramètres statiques sont définis une fois au début de l'exécution de l'algorithme et ne changent pas pendant toute la durée de la recherche. Ces paramètres peuvent inclure des éléments tels que :

- **Le nombre de gouttes d'eau (IWD)** : Le nombre total d'agents de recherche (gouttes d'eau) utilisés dans l'algorithme.
- **Les paramètres de mise à jour de la qualité du sol** : Les valeurs initiales et les coefficients utilisés pour la mise à jour des qualités du sol, qui influencent la manière dont les gouttes d'eau modifient leur environnement.
- **Les coefficients de déposition de la saleté** : Les valeurs qui déterminent la quantité de "saleté" déposée par chaque goutte d'eau, affectant la mémoire de l'algorithme sur les chemins empruntés.
- **Les paramètres d'évaporation** : Les taux d'évaporation qui régulent la suppression progressive des traces laissées par les gouttes d'eau.

Ces paramètres sont essentiels pour définir le comportement de base de l'algorithme et restent constants pour garantir une structure stable et cohérente tout au long de la recherche de la solution optimale.

Phase 2 : Initialisation des paramètres dynamiques

Cette phase consiste à initialiser les paramètres dynamiques, c'est-à-dire ceux qui changent au cours du processus de recherche. Voici une explication détaillée de cette phrase : Les paramètres dynamiques sont définis et initialisés au début de la construction de la solution, ce qui signifie qu'ils commencent avec des valeurs spécifiques avant que le processus de création de la solution ne commence. Pendant le processus de construction de la solution, ces paramètres sont continuellement mis à jour en fonction des étapes et des décisions prises par l'algorithme. À la fin de chaque itération, c'est-à-dire après qu'une solution complète a été construite et évaluée, ces paramètres dynamiques sont réinitialisés à leurs valeurs initiales avant le début de la prochaine itération. Cela permet à chaque itération de recommencer avec un état propre et assure que les ajustements faits dans une itération ne biaisent pas les suivantes.

Phase 3 : Répartition d'un créneau horaire aléatoire sur un examen aléatoire pour tous les IWDs

Cette phase consiste à initialiser la recherche des solutions en attribuant de manière aléatoire un créneau horaire à un examen pour chaque agent de recherche (IWD). Voici une explication détaillée :

Dans cette phase, l'algorithme Adapted IWD commence par assigner de manière aléatoire un créneau horaire à un examen spécifique pour chaque goutte d'eau intelligente (IWD). Concrètement :

1. **Sélection d'un examen aléatoire** : Pour chaque IWD, un examen est choisi aléatoirement parmi l'ensemble des examens disponibles.
1. **Attribution d'un créneau horaire aléatoire** : Ensuite, un créneau horaire est attribué de manière aléatoire à cet examen sélectionné.

Ce processus détermine le premier point de départ ou le premier examen visité par chaque IWD dans sa quête pour construire une solution optimale. En assignant ces points de départ aléatoirement, l'algorithme assure une diversité initiale dans les chemins explorés par les différents IWDs, ce qui aide à couvrir une plus grande partie de l'espace de recherche et augmente les chances de trouver des solutions optimales en évitant les biais initiaux.

Phase 4 : Construction des solutions

Cette phase vise à élaborer des solutions complètes en utilisant l'algorithme IWD. Voici une explication détaillée :

Objectif : Construire des solutions complètes de planification des examens en mettant à jour constamment le tableau des examens visités par chaque goutte d'eau intelligente (IWD).

Processus :

1. **Mise à jour initiale** : Un tableau indiquant quels examens ont été visités par chaque IWD est mis à jour en continu.
2. **Construction incrémentale** : À chaque étape de cette phase, chaque IWD étend sa solution partielle en ajoutant un nouvel examen à son emploi du temps. Cet ajout doit être un composant réalisable, c'est-à-dire qu'il ne doit violer aucune contrainte dure du problème (comme des conflits d'examens).

Boucles de construction :

- **Boucle externe** : Dans cette boucle, chaque examen doit se voir attribuer un créneau horaire. Cette boucle assure que tous les examens sont programmés.
- **Boucle interne** : Pour chaque IWD, la solution est construite examen par examen. L'algorithme utilise le degré de saturation (SD) pour déterminer le meilleur créneau horaire pour chaque examen. Le degré de saturation est une mesure qui aide à choisir l'examen et le créneau horaire qui minimiseront les conflits futurs.

Transition à travers le graphe :

- Les IWDs traversent un graphe où chaque nœud représente un examen et chaque arête représente la possibilité d'attribuer un examen à un créneau horaire spécifique.
- La construction se poursuit jusqu'à ce que toutes les solutions soient complètes, c'est-à-dire que chaque IWD ait attribué un créneau horaire à tous les examens.

Critères d'arrêt :

- La phase se termine lorsque les critères d'arrêt définis pour la construction de la population complète sont remplis. Ces critères peuvent inclure la complétion d'un certain nombre d'itérations ou l'atteinte d'un niveau de qualité de solution prédéfini.

En résumé, cette phase construit des solutions complètes en attribuant des créneaux horaires à chaque examen tout en respectant les contraintes du problème, en utilisant des boucles imbriquées et des techniques comme le degré de saturation pour optimiser le processus.

Phase 5 : Amélioration des solutions

Cette phase se concentre sur l'amélioration des solutions initiales qui ont été construites lors de la phase précédente. Voici une explication détaillée de cette phase :

Objectif :

- Améliorer la qualité des solutions générées pendant la phase de construction.
- Réduire les conflits et optimiser la répartition des examens dans les créneaux horaires.

Processus :

1. **Évaluation des solutions** : Chaque solution construite est évaluée pour identifier les points faibles, tels que les conflits entre examens ou une répartition inefficace des créneaux horaires.
2. **Optimisation locale** :
 - **Recherche locale** : Un optimiseur de recherche locale est appliqué à chaque solution pour explorer les environs immédiats de la solution actuelle et trouver des améliorations. Cela peut inclure des techniques comme le déplacement d'examens, l'échange de créneaux horaires, ou la réaffectation des examens.
 - **Structures de voisinage** : Utilisation de structures de voisinage efficaces pour guider la recherche locale. Ces structures définissent quels mouvements ou modifications peuvent être faits à la solution actuelle pour améliorer sa qualité.
3. **Critères de choix des améliorations** :
 - Les modifications apportées à la solution doivent respecter les contraintes du problème, notamment en évitant les conflits d'examens.
 - Chaque modification doit conduire à une amélioration mesurable de la solution, souvent évaluée par une fonction de coût ou une métrique de qualité.
4. **Itérations de l'amélioration** :
 - Le processus d'amélioration est itératif, chaque solution étant modifiée plusieurs fois jusqu'à ce qu'aucune amélioration significative ne puisse être trouvée.
 - Les critères d'arrêt pour l'amélioration peuvent inclure un nombre maximal d'itérations, une amélioration minimale atteinte, ou le temps de calcul écoulé.
5. **Convergence** :
 - Les solutions sont continuellement améliorées jusqu'à ce qu'elles atteignent un état où elles ne peuvent plus être facilement améliorées, indiquant que l'algorithme a convergé vers une solution optimale ou quasi-optimale.

Bénéfices :

- Cette phase permet de transformer les solutions initiales en solutions de haute qualité, optimisant la répartition des examens et minimisant les conflits.
- Assure que les solutions finales sont robustes et efficaces, répondant aux exigences du problème de planification des examens.

En résumé, la phase d'amélioration des solutions se concentre sur l'optimisation des solutions générées précédemment, en utilisant des techniques de recherche locale pour affiner et perfectionner les emplois du temps des examens, tout en respectant les contraintes du problème.

Phase 6 : Résiliation

Cette phase conclut l'exécution de l'algorithme en déterminant quand arrêter les itérations des phases de construction et d'amélioration des solutions. Voici une explication détaillée de cette phase :

Objectif :

- Définir les conditions qui déterminent quand l'algorithme doit s'arrêter, garantissant que des solutions optimales ou satisfaisantes sont trouvées sans effectuer de calculs inutiles.

Processus :

1. Conditions de terminaison :

- Les phases de construction (Phase 4) et d'amélioration (Phase 5) des solutions sont répétées jusqu'à ce qu'une condition de terminaison soit atteinte.
- Les conditions de terminaison peuvent inclure :
 - Un nombre maximal d'itérations.
 - Un temps de calcul maximal.
 - Une amélioration minimale entre les itérations (convergence).

2. Adaptation et amélioration :

- Les étapes mentionnées précédemment sont réalisées par un algorithme IWD adapté. Cependant, les chercheurs ont aussi modifié cet algorithme pour améliorer son efficacité.
- **Concept de Global Best :**
 - Les chercheurs ont introduit le concept de "Global Best" pour diversifier les solutions et éviter de se piéger dans des optima locaux.
 - En utilisant le Global Best, l'algorithme favorise les solutions qui sont les meilleures globalement parmi toutes les itérations, permettant ainsi une exploration plus large de l'espace de recherche.
 - Cela aide l'algorithme à se diriger vers des régions prometteuses de l'espace de solutions, améliorant les chances de trouver des solutions optimales.

Bénéfices :

- L'utilisation du Global Best permet à l'algorithme de maintenir un bon équilibre entre exploration et exploitation, augmentant ainsi la diversité des solutions et améliorant la qualité globale.
- La phase de résiliation assure que les ressources de calcul sont utilisées efficacement et que l'algorithme s'arrête une fois qu'il est peu probable de trouver de meilleures solutions.

Résumé : En résumé, la phase de résiliation consiste à déterminer quand arrêter les itérations des phases de construction et d'amélioration des solutions. En utilisant des conditions de terminaison claires et le concept de Global Best, l'algorithme IWD modifié parvient à diversifier les solutions et à éviter les optima locaux, permettant ainsi d'explorer des régions prometteuses de l'espace de recherche et d'améliorer les solutions trouvées.

Optimiseur de recherche locale (LSO)

Dans l'algorithme Hybrid IWD, la principale modification apportée à l'algorithme IWD modifié est l'intégration du Local Search Optimizer (LSO). Le LSO est introduit après la phase de construction des solutions, comme indiqué dans les lignes 24 à 26 de l'algorithme 1 . Cette intégration vise à équilibrer l'exploration et l'exploitation de l'espace de recherche, améliorant ainsi la qualité des solutions obtenues.

1. Initialize static parameters.
2. **for** ($NI = 1$ to $iter_{max}$) **do**
3. Initialize dynamic parameters.
4. Spread one random timeslot (Tslot) to a random exam for all IWD_s .
5. Update the list of visited node $V_s(IWD)$.
6. **for** ($e = 1$ to $E - 1$) **do**
7. **for** ($s = 1$ to Drops) **do**
8. Determine next exam n_e to be scheduled in the solution IWD_s using saturation degree.
9. **If** $IWD.Exam.Tslot(s, n_e, T^{Tb}(n_e)) = 1$ **then**
10. $IWD_s(n_e) = T^{Tb}(n_e)$
11. **Else**
12. **for** ($t = 0$ to P)
13. **If** $IWD.Exam.Tslot(s, n_e, p) = 1$ **then**
14. Push Tslot t into vector ϕ .
15. **Endif**
16. **Endfor**
17. $\tau =$ select random timeslot from ϕ
18. $IWD_s(n_e) = \tau$
19. **Endif**
20. update Velocity of the solution IWD_s (Eq. 7)

21. update Soil of the solution IWD_s (Eq. 8,9)
22. **Endfor**
23. **Endfor** (Eq. 1,2,3,4,5,6,7,8,9,10)
24. **If** $RAND(0,1) < LSR$
25. $IWD_s = LSO()$
26. **Endif**
27. update the soil value of all edges included in the T^{Tb} (Eq. 12)
28. update the global best solution T^{Tb} (Eq. 13)
29. **if** value of $T^{Tb} > T^{lb}$ **then**
30. $T^{Tb} = T^{lb}$
31. **Endif** (Eq. 14)
32. **Endfor**
33. Stops with the global best solution.

Algorithme 1 : Les principales étapes de l'algorithme Hybrid IWD proposé

Explication de l'Algorithme 1: Algorithme Hybrid IWD

L'algorithme 1 montre les étapes principales pour l'algorithme IWD hybride proposé. Voici une explication détaillée des étapes de cet algorithme :

1. **Initialiser les paramètres statiques :**
 - Cette étape configure les paramètres fixes de l'algorithme, qui ne changent pas au cours de l'exécution.
2. **Boucle principale (N) :**
 - **Début :** Répéter jusqu'à ce que le nombre maximum d'itérations soit atteint.
 - **Initialiser les paramètres dynamiques :** Réinitialiser les paramètres qui changent au cours de chaque itération de l'algorithme.
 - **Distribuer un créneau horaire aléatoire (Tslot) :** Attribuer un créneau horaire aléatoire pour chaque IWD (fourmi).
 - **Mettre à jour la liste des nœuds visités ($V_{i(IWD)}$) :** Enregistrer les créneaux horaires visités par chaque IWD.
3. **Boucle de construction des solutions (t) :**
 - **Début :** Répéter jusqu'à ce que le nombre maximum de gouttes soit atteint.
 - **Déterminer le prochain examen à planifier :** Utiliser le degré de saturation pour choisir le prochain examen à planifier.
 - **Condition de planification :** Si l'examen peut être planifié dans le créneau horaire sélectionné, alors :
 - Mettre à jour la liste des nœuds visités et le créneau horaire correspondant.
 - **Sinon :** Sélectionner un créneau horaire aléatoire pour l'examen.
 - **Mettre à jour la vélocité :** Adapter la vitesse de l'IWD .
 - **Mettre à jour le sol :** Ajuster la quantité de sol transportée par l'IWD.
4. **Boucle de vérification du taux de recherche locale (LSR) :**
 - **Si $RAND(0,1) < LSR$:** Invoquer l'optimiseur de recherche locale (LSO) avec une probabilité égale à LSR.
 - **LSO :** Améliorer localement la solution jusqu'à ce que le nombre maximum d'itérations soit atteint.
5. **Mettre à jour le sol sur tous les arêtes :** Ajuster la quantité de sol sur les arêtes visitées.
6. **Mettre à jour la meilleure solution globale :**
 - Si la nouvelle solution est meilleure que la solution globale actuelle, alors mettre à jour la solution globale.
7. **Fin de la boucle principale (N) :** Terminer si les conditions d'arrêt sont remplies.

8. **Terminer avec la meilleure solution globale** : Arrêter l'algorithme avec la meilleure solution trouvée.

Concepts Clés

- **LSR (Local Search Rate)** : Détermine la probabilité d'invocation du LSO. Une valeur élevée de LSR signifie une exploitation plus intensive.
- **Vélocité et Sol** : Paramètres dynamiques qui influencent le comportement des IWD lors de la recherche de solutions optimales.
- **Optimisation Locale** : Utilisation du LSO pour améliorer la qualité des solutions en affinant les solutions partiellement optimales.

En résumé, l'algorithme Hybrid IWD combine la recherche globale par les IWD avec l'optimisation locale fournie par le LSO pour obtenir des solutions de haute qualité pour le problème de planification des examens.

L'optimiseur de recherche locale utilise une structure de voisinage, adaptée dans cette version de l'IWD, pour générer un nouvel ensemble de solutions voisines (voir l'algorithme 2). Les descriptions détaillées de la structure du voisinage et du processus de génération de solutions voisines sont présentées dans la sous-section suivante. La recherche locale ou recherche de voisinage est un cadre méta heuristique très efficace pour résoudre un grand nombre de problèmes d'optimisation et de satisfaction de contraintes. Les algorithmes de recherche locale tentent d'améliorer progressivement la solution courante en explorant les solutions voisines. Ainsi, la solution courante est itérativement remplacée par une solution voisine si celle-ci est réalisable et présente une meilleure valeur de fitness.

L'un des principaux avantages de la recherche locale réside dans la méthode de description du voisinage. En général, une bonne méthode de voisinage permet une recherche efficace et conduit à d'excellents résultats, principalement basés sur la solution initiale. Les faibles performances de recherche de voisinage sont étroitement liées à la qualité de la solution initiale. De plus, le comportement de la recherche locale dépend fortement des caractéristiques du voisinage. Par exemple, certains voisinages permettent des améliorations significatives et rapides de la solution, mais seulement pour un nombre limité d'itérations. En revanche, d'autres voisinages n'autorisent que de petites améliorations, mais sur une plus longue période.

```
1. INPUT IWDs, f(IWDs)
2. DO
3. i = random integer number in range [0 - 2]
4. IF i = 0 THEN
5. IWDs = LSO - SingleMove(IWD's)
6. ELSEIF i = 1 THEN
7. IWDs = LSO - Swap(IWD's)
8. ELSE
9. IWDs = LSO - Kempe - Chain(IWD's)
10. ENDIF
11. IF f(IWD's) > f(IWDs) THEN
12. IWDs = IWD's
13. f(IWDs) = f(IWD's)
14. ENDIF
15. WHILE (The Itermax number is not met)
```

Algorithme 2 : algorithme d'optimisation de recherche locale (LSO) proposé.

Dans le contexte de PPEU, la structure de voisinage est utilisée pour générer de nouvelles solutions voisines en perturbant légèrement une solution existante. Chaque solution voisine est obtenue à partir de la solution de travail en utilisant un processus de mouvement, où des petites perturbations sont appliquées pour explorer le voisinage immédiat.

Pour améliorer l'efficacité de la recherche locale, il est possible de combiner plusieurs quartiers avec des caractéristiques complémentaires. Dans cette étude, trois quartiers sont étudiés : le mouvement (Move), l'échange (Swap), et la chaîne de Kempe (Kempechain). Ces quartiers sont utilisés à l'intérieur de l'optimiseur de recherche locale (LSO) pour explorer en profondeur la région de recherche de la solution en cours. En utilisant ces trois structures de quartier, le LSO peut découvrir une gamme étendue de solutions potentielles et optimiser la qualité de la solution finale.

IWD hybride pour PPEU

L'IWD hybride pour PPEU intègre le Local Search Optimizer (LSO) dans la boucle d'amélioration de l'IWD adapté en tant que nouvelle phase (phase 6). Cette intégration vise à renforcer la capacité d'exploitation de l'IWD adapté. Une fois la phase d'amélioration des solutions terminée, le LSO est activé avec une probabilité de taux de recherche local (LSR) compris entre 0 et 1. Cette probabilité détermine le pourcentage d'utilisation du LSO : plus la valeur de LSR est élevée, plus le LSO est fréquemment utilisé pour améliorer localement la qualité de chaque solution de planification, jusqu'à ce que les optima locaux soient atteints. En ajustant la valeur de LSR, on peut contrôler le niveau d'exploitation effectué par l'algorithme,

permettant ainsi d'équilibrer entre exploration et exploitation pour obtenir des solutions optimales.

Importance des algorithmes hybrides

Les algorithmes hybrides sont importants car ils parviennent à combiner les points forts de différentes techniques d'optimisation, offrant ainsi un équilibre entre l'exploration et l'exploitation. Les techniques basées sur la population, comme les algorithmes génétiques ou les algorithmes IWD, sont efficaces pour identifier des zones prometteuses dans l'espace de recherche, tandis que les techniques de recherche locale, comme la programmation quadratique séquentielle ou l'ILS, sont excellentes pour affiner et optimiser les solutions au sein de ces zones. L'hybridation permet donc de surmonter les limitations de chaque approche lorsqu'elles sont utilisées individuellement, résultant en des solutions plus robustes et plus efficaces[16].

Exemples et application dans divers domaines

Exemples d'algorithmes hybrides

1. IWD avec programmation quadratique séquentielle : Utilisé pour résoudre des problèmes complexes comme le problème de l'arbre de Steiner (STP), cette hybridation améliore la convergence et les performances de l'algorithme IWD original en optimisant les solutions locales à chaque itération.

2. IWD avec recherche locale itérée (ILS) : Appliqué à des problèmes de planification des commandes où certaines commandes peuvent être refusées. Cette combinaison permet de sélectionner les meilleures commandes tout en optimisant leur planification, ce qui diminue le nombre de commandes en retard et augmente les bénéfices du fabricant.

Applications dans divers domaines

1. Planification des examens universitaires (PPEU) : Les algorithmes hybrides peuvent optimiser l'attribution des examens à des créneaux horaires et des salles en respectant des contraintes strictes et souples, améliorant ainsi l'efficacité et la qualité du calendrier.

2. Gestion de la production industrielle : Dans la planification des commandes, les algorithmes hybrides aident à sélectionner les meilleures commandes à traiter et à planifier leur production, optimisant ainsi l'utilisation des ressources et maximisant les profits.

3. Routage et optimisation des réseaux : Les algorithmes hybrides, comme l'IWD combiné avec des techniques de recherche locale, sont utilisés pour résoudre des problèmes de routage

complexes, comme le problème de l'arbre de Steiner, en trouvant des solutions optimales plus rapidement et efficacement.

4. Logistique et transport : Les techniques hybrides peuvent être appliquées pour optimiser les itinéraires de livraison, minimiser les coûts et les délais, et améliorer l'efficacité générale des opérations logistiques.

En combinant différentes approches, les algorithmes hybrides offrent des solutions plus robustes et performantes pour une variété de problèmes complexes dans divers domaines.

5. conclusion

En conclusion, le chapitre sur l'algorithme hybride intelligent de gouttes d'eau pour le problème de calendrier d'examen a démontré l'efficacité et la pertinence de cette approche avancée pour la planification des examens universitaires. En combinant l'Algorithme Intelligent Water Drops (IWD) avec d'autres techniques d'optimisation, nous avons montré comment il est possible de créer des emplois du temps optimisés qui répondent aux contraintes et aux besoins spécifiques des institutions académiques. L'algorithme hybride proposé non seulement améliore la qualité des solutions obtenues, mais il offre également une flexibilité et une robustesse accrues face à des problèmes complexes de planification. Les performances prometteuses de cette approche soulignent son potentiel pour des applications futures dans divers domaines nécessitant une gestion efficace et adaptable des ressources et des horaires.

Chapitre III

Expérimentations et résultats

1. La base de données

La base de données Toronto Benchmark, introduite par MW Carter et al. (1996), est un ensemble de données largement utilisé pour évaluer les performances des algorithmes d'optimisation, en particulier dans le domaine de la programmation des examens. Cet ensemble de données, également appelé "Toronto Benchmark Suite," est composé de diverses études de cas du monde réel, fournissant des scénarios de programmation des examens tirés d'institutions académiques réelles.

Les ensembles de données de cette suite sont disponibles gratuitement et sont destinés à tester et comparer différents algorithmes de résolution de problèmes d'ordonnancement, comme l'algorithme hybride IWD (Intelligent Water Drops).

Cependant, il existe des versions différentes de ces ensembles de données, bien que portant les mêmes noms, ce qui a créé une certaine confusion dans la littérature. Ces versions se distinguent principalement par plusieurs caractéristiques :

- **Nombre d'étudiants** : Chaque version peut inclure un nombre différent d'étudiants participant aux examens.
- **Nombre d'inscriptions** : Le nombre total d'inscriptions aux examens peut varier.
- **Nombre d'examens** : Le nombre d'examens programmés peut être différent d'une version à l'autre.
- **Densité des conflits** : La densité des conflits, c'est-à-dire la fréquence des chevauchements ou des incompatibilités entre les examens, peut également varier.

R. Qu et al. (2009) ont classifié ces versions pour mieux comprendre et comparer les performances des algorithmes testés sur ces ensembles de données. En résumé, la base de données Toronto Benchmark est un outil essentiel pour les chercheurs en optimisation et programmation des examens, offrant des scénarios de test standardisés mais avec des variations significatives entre différentes versions.

Les métriques utilisées

Les ensembles de données de la Toronto Benchmark Suite sont souvent évalués à l'aide de plusieurs métriques clés qui permettent de comparer les performances des différents algorithmes d'optimisation pour la programmation des examens. Voici quelques-unes des métriques couramment utilisées :

1. **Nombre de créneaux horaires (Time Slots)** : Le nombre total de créneaux horaires utilisés pour programmer tous les examens. Une utilisation efficace des créneaux horaires est souvent souhaitée pour minimiser la durée totale de la période d'examen.
2. **Nombre de conflits (Conflicts)** : Le nombre total de conflits d'examen, c'est-à-dire le nombre de fois où des étudiants sont inscrits à deux examens programmés en même temps. Réduire ce nombre est crucial pour assurer une programmation sans chevauchement.
3. **Densité des conflits (ConflictDensity)** : La proportion de paires d'examens pour lesquelles il existe au moins un étudiant inscrit aux deux examens. Cela donne une idée de la difficulté du problème de programmation des examens, une densité plus élevée indiquant un problème plus difficile.
4. **Fonction de coût (CostFunction)** : Une métrique qui combine plusieurs critères pour évaluer la qualité globale d'un emploi du temps. Par exemple, une fonction de coût peut prendre en compte à la fois le nombre de conflits et la répartition des examens sur les créneaux horaires. Les fonctions de coût spécifiques peuvent varier en fonction des objectifs particuliers de l'algorithme ou de l'institution.
5. **Écart-type des écarts entre les examens (Standard Deviation of Gaps Between Exams)** : Cette métrique mesure la répartition des intervalles de temps entre les examens pour chaque étudiant. Un écart-type faible indique une répartition plus uniforme des écarts, ce qui est généralement souhaitable pour éviter les périodes de stress intense ou de longues attentes pour les étudiants.
6. **Pénalité (Penalty)** : Une mesure souvent utilisée dans les fonctions de coût, représentant les pénalités attribuées en fonction des préférences ou des contraintes spécifiques, telles que la distribution des examens pour éviter des périodes de surcharge pour les étudiants.

Ces métriques permettent d'évaluer de manière quantitative la performance des algorithmes de programmation des examens sur les ensembles de données de la Toronto Benchmark Suite, offrant ainsi un cadre standardisé pour la comparaison et l'amélioration des méthodes d'optimisation.

Les paramètres

Les paramètres sont les valeurs ajustées par l'algorithme pendant l'exécution afin d'optimiser la solution mentionnant :

velocity: Vitesse de chaque goutte d'eau.

Exemple d'initialisation: [1.0 for _ in range(drops)]

soil: Quantité de sol transportée par chaque goutte d'eau.

Exemple d'initialisation: [0.01 for _ in range(drops)]

best_solution: Meilleure solution trouvée au cours des itérations.

best_penalty: Pénalité associée à la meilleure solution trouvée

Les hypers paramètres

Les hyperparamètres sont des variables dont les valeurs sont définies avant le processus et qui contrôlent le comportement de l'algorithme. Ils ne sont pas modifiés par l'algorithme pendant l'exécution. Voici les hyperparamètres identifiés dans le code :

num_timeslots = 15 : nombre total de créneaux horaires

itermax = 100 : Nombre maximum d'itérations pour l'algorithme.

drops = 10 : Nombre de gouttes d'eau utilisées dans l'algorithme

lsr = 0.5 Probabilité d'appliquer une recherche locale à la solution courante

Optimisation et Contraintes :

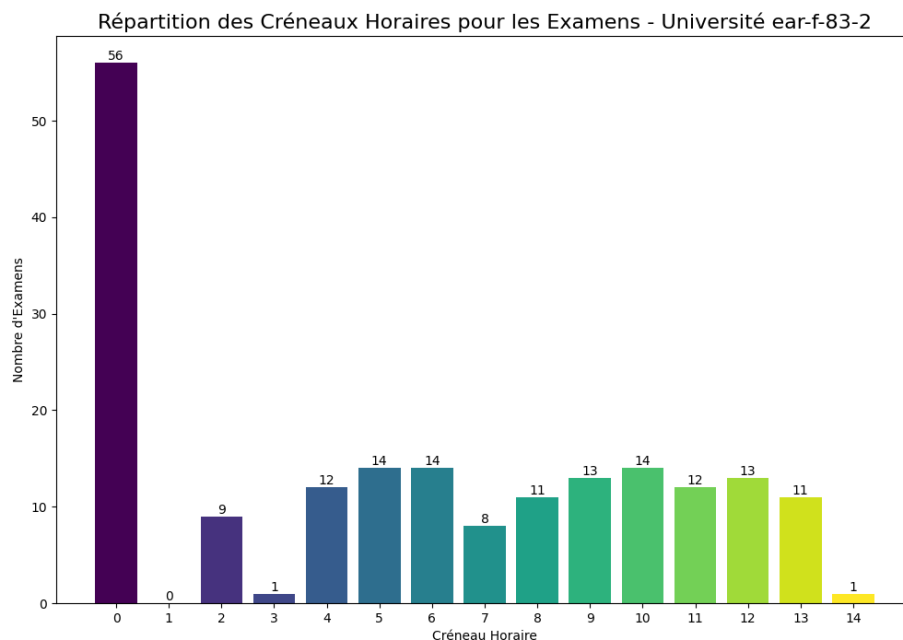
L'algorithme cherche à optimiser l'emploi du temps en respectant plusieurs contraintes :

- **Minimisation des conflits directs** : Éviter que des étudiants aient des examens en même temps.
- **Réduction des pénalités liées à la proximité temporelle** : Espacer les examens pour les étudiants de manière raisonnable.

2. les résultats d'expérimentation

Le graphique à barres montre la distribution des examens sur les différents créneaux horaires de 0 à 14 pour chaque universités (fig 1 fig2..... 4). Chaque barre représente un créneau horaire, et la hauteur de la barre représente le nombre d'examens attribués à ce créneau.

- **Créneaux Chargés :** Les créneaux horaires avec des barres plus hautes ont plus d'examens programmés.
- **Créneaux Moins Chargés :** Les créneaux horaires avec des barres plus basses ou inexistantes ont moins ou aucun examen programmé.



Répartition des Créneaux Horaires pour les Examens - université ear-f-83-2

Figure 1 : Cette figure montre un histogramme intitulé "Répartition des Créneaux Horaires pour les Examens - université ear-f-83-2". Voici une explication générale des résultats affichés

Interprétation du Graphique

- **Créneau horaire 0 :** Le nombre d'examens est extrêmement élevé, avec un total de 56 examens. Cela indique une concentration excessive d'examens dans ce créneau.
- **Autres créneaux horaires :** Les créneaux horaires 1 et 3 sont presque vides, avec 0 et 1 examen respectivement.
- **Créneau horaire 2 :** Modérément utilisé avec 9 examens.

- **Créneaux horaires 4 à 12** : Relativement bien distribués avec un nombre d'examens variant entre 8 et 14, ce qui montre une répartition plus équilibrée des examens dans ces créneaux.
- **Créneau horaire 13** : Légèrement moins utilisé avec 11 examens.
- **Créneau horaire 14** : Très peu utilisé avec seulement 1 examen.

Analyse des Résultats

1. Surutilisation du Créneau Horaire 0 :

- Il y a une concentration excessive d'examens dans le créneau horaire 0, ce qui pourrait entraîner des conflits importants pour les étudiants et les ressources disponibles (salles, surveillants, etc.).
- Cela suggère que l'algorithme utilisé pour la planification n'a pas réussi à répartir uniformément les examens ou qu'il y a des contraintes spécifiques favorisant ce créneau.

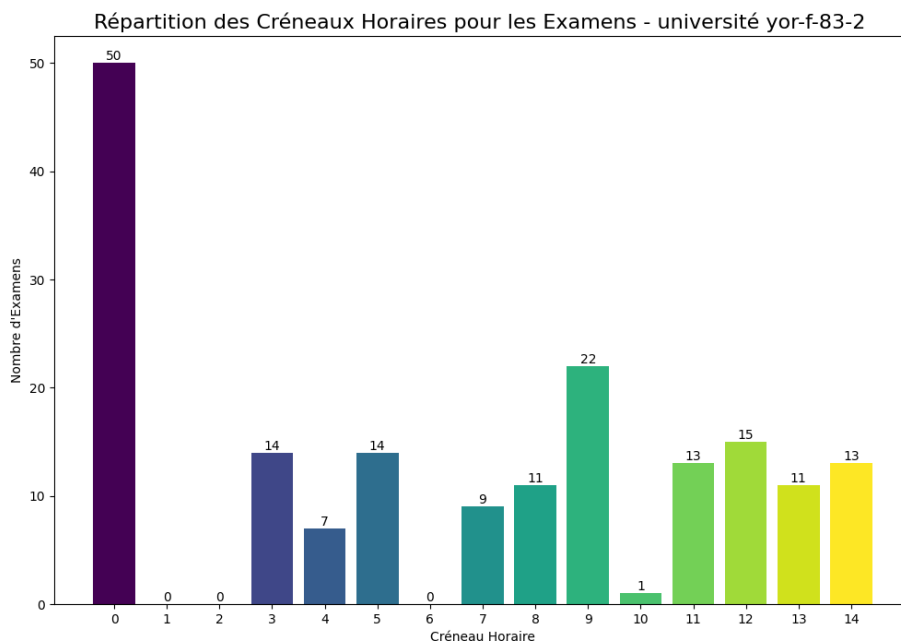
2. Sous-utilisation des Créneaux Horaires 1, 3 et 14 :

- Ces créneaux sont presque vides, indiquant une mauvaise utilisation des ressources disponibles durant ces périodes.
- Cela peut être dû à des contraintes externes (comme la disponibilité des salles ou des enseignants) ou à des inefficacités dans l'algorithme de planification.

3. Distribution Équilibrée dans les Créneaux 4 à 12 :

- Ces créneaux montrent une répartition plus équilibrée et uniforme des examens, ce qui est un signe positif.
- Une telle répartition aide à réduire les conflits et à optimiser l'utilisation des ressources académiques.

Ces résultats montrent que bien que certaines périodes soient bien équilibrées, il y a des points critiques à aborder pour améliorer l'efficacité et l'équité de la planification des examens.



Répartition des Créneaux Horaires pour les Examens - université yor-f-83-2

Figure 2 : Cette figure montre un histogramme intitulé "Répartition des Créneaux Horaires pour les Examens - université yor-f-83-2". Voici une explication générale des résultats affichés :

1. Distribution :

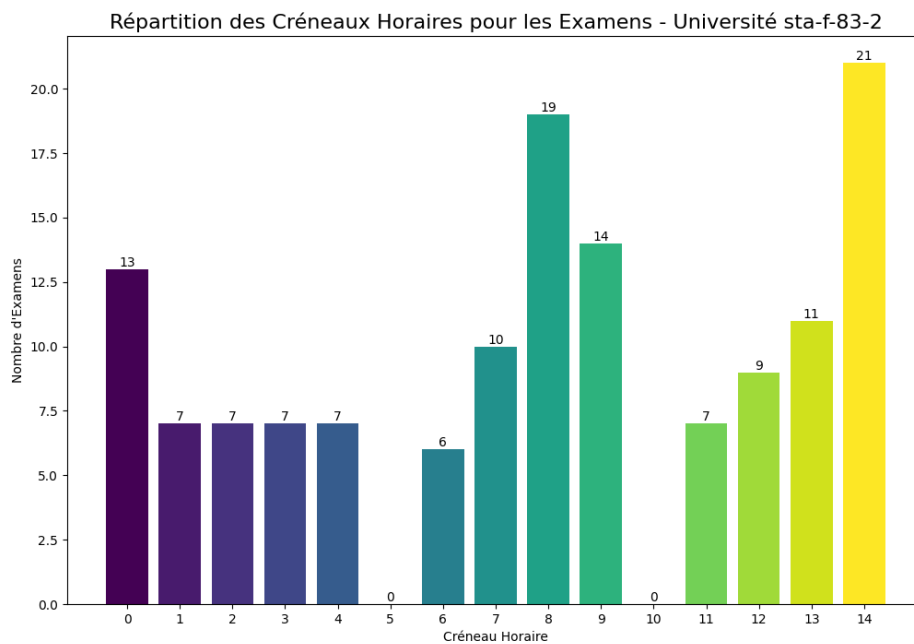
- Le créneau horaire 0 a le plus grand nombre d'examens avec 50, ce qui est significativement plus élevé que les autres créneaux.
- Les créneaux horaires 1, 2 et 6 n'ont aucun examen planifié.
- Les créneaux horaires 3 et 5 ont chacun 14 examens.
- Le créneau horaire 4 a 7 examens.
- Le créneau horaire 7 a 9 examens.
- Le créneau horaire 8 a 11 examens.
- Le créneau horaire 9 a 22 examens, le deuxième plus grand nombre après le créneau 0.
- Le créneau horaire 10 a 1 examen.
- Les créneaux horaires 11 et 14 ont chacun 13 examens.
- Le créneau horaire 12 a 15 examens.
- Le créneau horaire 13 a 11 examens.

2. Observations :

- Le créneau horaire 0 est exceptionnellement chargé avec 50 examens, ce qui peut indiquer une forte concentration d'examens à ce moment-là.

- Les créneaux horaires 1, 2 et 6 sont complètement vides, sans aucun examen planifié.
- Le créneau horaire 9 est également assez chargé avec 22 examens.
- La répartition des examens semble inégale, avec certains créneaux ayant beaucoup plus d'examens que d'autres.

En résumé, l'histogramme montre une répartition très inégale des examens sur les différents créneaux horaires, avec une concentration extrêmement élevée au créneau 0 et une absence totale d'examens aux créneaux 1, 2 et 6. D'autres créneaux, comme 8 et 11, montrent également des nombres relativement élevés d'examens, tandis que la plupart des autres créneaux ont des nombres plus modestes.



Répartition des Créneaux Horaires pour les Examens - Université sta-f-83-2

Figure 3 : Cette figure montre un histogramme intitulé "Répartition des Créneaux Horaires pour les Examens - Université sta-f-83-2". Voici une explication générale des résultats affichés

1. Distribution :

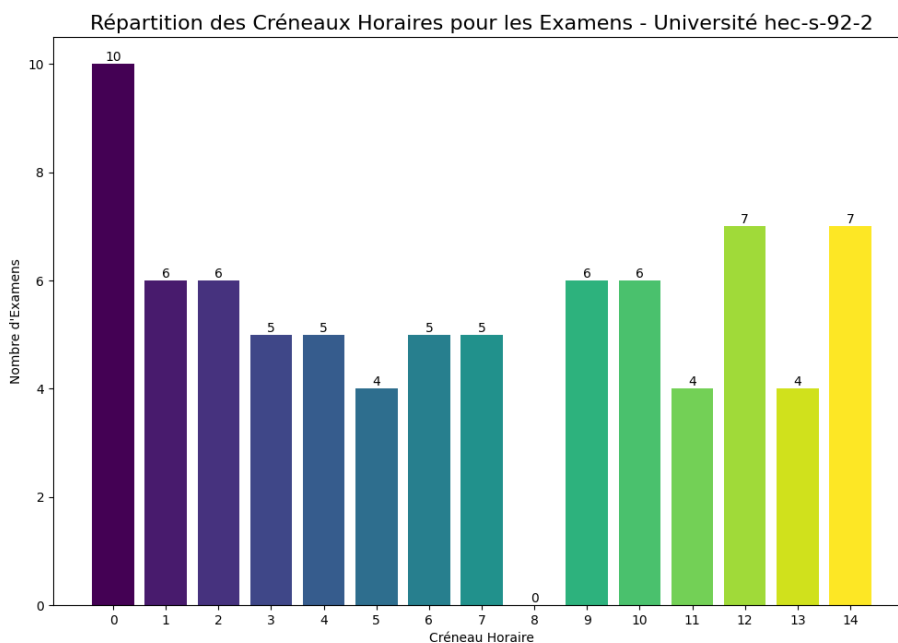
- Le créneau horaire 0 a 13 examens.
- Les créneaux horaires 1, 2, 3, et 4 ont chacun 7 examens.
- Le créneau horaire 6 a 6 examens, tandis que le créneau 5 n'en a aucun.
- Le créneau horaire 7 a 10 examens.

- Le créneau horaire 8 a le nombre le plus élevé avec 19 examens, suivi par le créneau horaire 9 avec 14 examens.
- Le créneau horaire 10 n'a pas d'examens planifiés.
- Les créneaux horaires 11 et 12 ont respectivement 7 et 9 examens.
- Le créneau horaire 13 a 11 examens, et le créneau horaire 14 a le deuxième plus grand nombre avec 21 examens.

2. Observations :

- Les créneaux horaires avec le plus d'examens sont les créneaux 8 et 14, avec respectivement 19 et 21 examens.
- Les créneaux 5 et 10 n'ont aucun examen planifié.
- Il y a une répartition variée des examens entre les créneaux horaires, avec certains créneaux (comme 0, 8, et 14) ayant beaucoup plus d'examens que d'autres.

En résumé, l'histogramme montre une répartition inégale des examens sur les différents créneaux horaires, avec des pics notables aux créneaux horaires 8 et 14, et des créneaux sans examens (5 et 10).



Répartition des Créneaux Horaires pour les Examens - Université hec-s-92-2

Figure 4 : Cette figure montre un histogramme intitulé "Répartition des Créneaux Horaires pour les Examens - Université hec-s-92-2". Voici une explication générale des résultats affichés

1. Distribution :

- Le créneau horaire 0 a le plus grand nombre d'examens, avec 10 examens.
- Les créneaux horaires 1 et 2 ont chacun 6 examens.
- Les créneaux horaires 3, 4, 6, et 7 ont chacun 5 examens.
- Le créneau horaire 5 a 4 examens.
- Le créneau horaire 8 n'a aucun examen (valeur de 0).
- Les créneaux horaires 9 et 10 ont chacun 6 examens.
- Le créneau horaire 11 a 4 examens.
- Les créneaux horaires 12 et 14 ont chacun 7 examens.
- Le créneau horaire 13 a 4 examens.

2. Observations :

- Le créneau horaire 0 est le plus chargé avec le plus grand nombre d'examens (10 examens).
- Le créneau horaire 8 est le seul créneau sans aucun examen.
- La distribution des examens est relativement équilibrée entre les autres créneaux horaires, avec une légère concentration autour de 4 à 7 examens pour la majorité des créneaux.

En résumé, l'histogramme montre que le créneau horaire 0 est le plus sollicité pour les examens à l'Université hec-s-92-2, tandis que le créneau horaire 8 est complètement libre. La répartition des examens est assez uniforme pour les autres créneaux, avec un nombre d'examens variant entre 4 et 7.

Les résultats de pénalité pour chaque université sont représentés ci-dessus dans le graphique à barres.

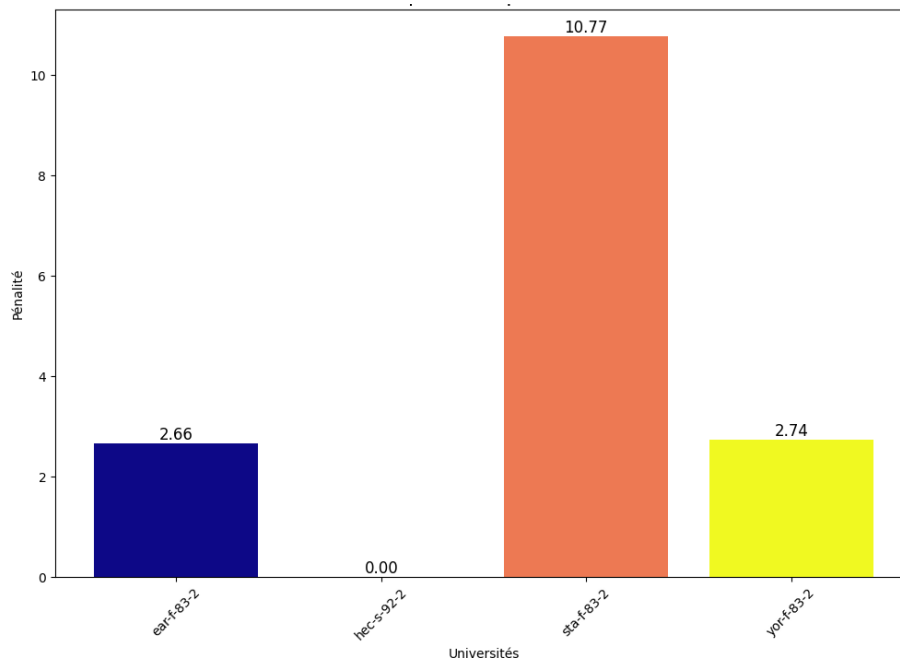


Figure 5: pénalité pour chaque Université

- **ear-f-83-2: Pénalité de 2.6624**
 - Cette pénalité est relativement faible, ce qui indique que la répartition des créneaux horaires est assez équilibrée et qu'il y a peu de conflits ou d'examens rapprochés pour les étudiants.
- **hec-s-92-2: Pénalité de 0.0**
 - Une pénalité de 0 signifie que la répartition des examens est optimale pour cette université. Il n'y a pas de conflits, et les étudiants ont suffisamment de temps entre les examens.
- **sta-f-83-2: Pénalité de 10.7786**
 - Cette pénalité est la plus élevée parmi les universités listées. Cela suggère que la répartition des créneaux horaires a des conflits significatifs, ce qui pourrait causer des difficultés pour les étudiants, comme des examens consécutifs ou des temps de révision insuffisants.
- **yor-f-83-2: Pénalité de 2.74**
 - La pénalité est faible. Cela indique une répartition des examens raisonnablement bien équilibrée avec quelques conflits possibles.

3. Une discussion

Les pénalités sont un indicateur de l'efficacité de la répartition des créneaux horaires pour les examens dans chaque université. Une pénalité plus faible signifie une meilleure répartition, minimisant les conflits et permettant aux étudiants de mieux se préparer entre les examens. À l'inverse, une pénalité élevée suggère des problèmes dans la répartition, qui peuvent rendre la période d'examen plus stressante pour les étudiants.

Pour ear-f-83-3 et yor-f-83-2 Les valeur de pénalité représente la pénalité totale associée à cet emploi du temps. La pénalité est une mesure de la "qualité" de l'emploi du temps en termes de confort pour les étudiants, c'est-à-dire la minimisation des conflits et de la proximité des examens pour les étudiants ayant plusieurs examens.

Les résultats obtenus montrent un emploi du temps optimisé pour 189 examens avec des pénalité modérée de 2.6624 et 2.74. Cela indique que l'algorithme a réussi à minimiser les conflits à optimiser la répartition des examens pour le confort des étudiants.

Pour hec-s-92-2 avec une pénalité de 0.0 est un exemple idéal de répartition des créneaux horaires, tandis que sta-f-83-2 avec une pénalité de 10.77 peut avoir des problèmes de conflit c'est a dire il peut y avoir des étudiants aient deux examens en même temps plus précisément dans le même créneau horaire.

Conclusion Générale

Conclusion générale

En conclusion, ce mémoire a démontré que l'optimisation des emplois du temps des examens universitaires grâce au génie logiciel et à l'intelligence artificielle, en particulier avec l'algorithme Intelligent Water Drops (IWD), représente une avancée significative dans la gestion académique. L'application de l'IWD a permis de générer des emplois du temps optimisés qui respectent les contraintes institutionnelles tout en maximisant l'utilisation des ressources et en améliorant la satisfaction des enseignants et des étudiants. Les résultats obtenus montrent une réduction des conflits et une meilleure adaptation aux imprévus, offrant une solution pratique, efficace et flexible pour les institutions académiques. Cette approche pave la voie à des futures intégrations d'apprentissage automatique et de méthodes hybrides pour renforcer encore plus la robustesse et l'efficacité des solutions proposées.

Références Bibliographiques

Références bibliographiques

- [1] : Zouambia Adel et Amazouz Assil « Conception et Réalisation d'un système interactif D'aide à La gestion des emplois du temps », ESI (Ex INI) Alger, Promotion 2009/2010. (06.07)
- [2] : Rédha Mili « Conception et implémentation d'un algorithme d'optimisation pour la Génération automatique d'emploi du temps Universitaire», Université 20 Août 1955 Skikda, Promotion 2010. (08.09)
- [3] : gestion des flux http://ressources.auneg.fr/nuxeo/site/esupversions/d746d9f6-9e2f-427a-bf2b-b03a9adcd99c/lorr_casanova_gestion_flux/co/2_2_2.html
- [4] : https://fr.wikipedia.org/wiki/Algorithme_de_Johnson
- [5] : Alignement optimal et comparaison de séquences génomiques et protéiques
<https://interstices.info/alignement-optimal-et-comparaison-de-sequences-genomiques-et-proteiques/>
- [6] : Introduction à la Programmation Linéaire en Nombres Entiers L. Wolsey, Integer Programming, John Wiley & Sons, Inc, New York, 1998.ogrammatio
- [7] : la programmation linéaire en mathématique
<https://www.eurodecision.com/algorithmes/recherche-operationnelle-optimisation/programmation-mathematique-programmation-lineaire>
- [8] : Programmation Linéaire en Nombres Entiers <https://dept-info.labri.fr/~robson/RO/cours9.pdf>
- [9] : Apprentissage supervisé : définition et exemples d'algorithme Antoine Crochet-Damais JDN
- [10] : <https://fr.indeed.com/m/jobs?q=Expert+System>
- [11] : DÉVELOPPEMENT D' UN SYSTEME EXPERT POUR LE RAISONNEMENT LOGIQUE TIJILLET 2018 p44 49
- [12] : benkorich redha, chellghouf abderrahmane , 'automatisation des Emplois du Temps ',2006 , (40,41,42,43)
- [13] : https://fr.wikipedia.org/wiki/Agent_intelligent
- [12] : <https://botpress.com/fr/blog/quels-sont-les-differents-types-d-agents-de-l-intelligence-artificielle>
- [13] : <https://www.amazon.fr/Architecture-hybride-comportement-temp%C3%A9ramental-Rationnel/dp/3841632815>
- [14] : http://www.monlyceenumerique.fr/nsi_premiere/algo_a/a6_algo_gloutons.php
- [15] : https://fr.wikipedia.org/wiki/Algorithme_g%C3%A9n%C3%A9tique
- [16] : Hybrid intelligent water Drops algorithm for examinationtimetablingproblem Volume 34, Issue 8, Part A, September 2022, Pages 4847-4859 <https://doi.org/10.1016/j.jksuci.2021.06.016>

[17] : An ensemble of intelligent water drop algorithm for feature selection optimization problem
Volume 65, April 2018, Pages 531-541 <https://doi.org/10.1016/j.asoc.2018.02.003> April 2018, Pages 531-541

[Matthieu, 2008] : Matthieu Aubry, Jeremie Guidoux, Quentin Marguet, Mathieu Vénisse-Garson,
«Gestion Emploi du temps» Université de Nantes, Novembre 2008.

[Troudi, 2006] : Mme Troudi Fatiha «Résolution du problème de l'emploi du temps: Proposition d'un
algorithme évolutionnaire multi objectif », Université Mentouri – Constantine, 2005-2006

[Leite et coll., 2018] : Un algorithme mémétique basé sur MOEA/D pour le problème de planification de
examens Doux. *Calculer.*, 22 (5) (2018), pages 1511 à 1523 , 10.1007/s00500-017-2886-y

[Leite et coll., 2019] : N. Leite , F. Melício , AC Rosa Un algorithme de recuit simulé rapide
pour le problème de planification des examens *Système expert. Appl.*, 122 (2019)
, pages 137 à 151 , 10.1016/j.eswa.2018.12.048

[Pillay et Banzhaf, 2009] : N. Pillay, W. Banzhaf Une étude des combinaisons heuristiques pour les systè
mes hyper-heuristiques pour le problème de planification des examens sans capacité EUR. *J. Opér. Rés.*,
197 (2)(2009), pages 482 à 491 , 10.1016/j.ejor.2008.07.023

[Aldeeb et coll., 2019] : BA Aldeeb , MA Al-Betar , AO Abdelmajeed , MJ Younes , M.
AlKenani, W. Alomoush, MA Alqahtani Un
examen complet du problème de calendrier des examens universitaires sans capacité *Int. J. Appl. Ing.*
Rés., 14 (24)(2019), pages 44 à 50

[Shah-Hosseini, 2012a] : H. Shah-Hosseini Une approche d'optimisation continue par
l'algorithme Intelligent Water Drops *Procedia – Soc. Comportement. Sci.*, 32 (2012), pages 100 à 110