REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE MATHEMATIQUES ET DE L'INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE

N° :……..

DOMAINE : Informatique

FILIERE : Informatique

OPTION : Informatique Décisionnel Optimisé

Mémoire présenté pour l'obtention

Du diplôme de Master Académique

Par: Gahgah Hadjer

Intitulé

Problème d'emploi de temps : proposition un algorithme bio-inspiré

Soutenu devant le jury composé de :

M.ABACH FARID	Université de M'sila Président
M. DABBA Ali	Université de M'sila Rapporteur
M. BENAZI MAKHELOUF	Université de M'sila Co-Rapporteur
M. GERNA ABD RAHIM	Université de M'sila Examinateur

Année universitaire : 2017 /2018

Remerciements

On remercie Dieu le tout puissant, qui nous a donné la force et la patience pour l'accomplissement de ce travail.

Nous remercions en particulier ***M.DABBA ALI****, pour l'honneur qu'il nous a fait de bien vouloir nous encadrer, et pour les conseils donnés lors de la réalisation de ce travail.*

On adresse nos remerciements au membre de jury pour avoir accepté de nous prêter de leur attention et évaluer notre travail.

Je réserve les derniers remerciements à toute ma famille et mes amis, en particulier ma mère pour ses sacrifices, ses encouragements, sa présence et son soutien au quotidien.

Dédicaces

Je dédie ce travail à …

Mes chers parents, que nulle dédicace ne peut exprimer mes sincères
sentiments, pour leur patience illimitée
leur encouragement contenu, leur aide, en témoignage de mon profond amour et respect pour ses grands sacrifices.

Mon cher frère FOUAD et ma chère sœur BASMA
Pour leur grand amour et leur soutien qu'ils trouvent ici
l'expression de ma haute gratitude.

Mes chers amies qui sans leur encouragement ce travail
n'auras jamais vu le jour.

Ma chère grande mère (Saadia)

Et à toute ma famille et à tous ceux que j'aime

Résumé

Cette dissertation offre les résultats d'une enquête, et l'optimisation du système de planification à l'Université de Msila (en tant que étude de cas) en utilisant l'algorithme génétique (GA). Les Techniques GA sont utiles pour résoudre le problème d'ordonnancement du monde réel tel que le calendrier qui est un travail complexe et généralement fait manuellement.

Ce travail se concentre sur l'horaire des cours avec attribution des événements (temps, sujet et conférencier) dans une manière appropriée en utilisant la ressource disponible et aide à éviter les conflits. Les algorithmes explorent des différents opérateurs de GA comme le croisement, la mutation et le mécanisme de sélection qui est appliqué à l'ensemble des chromosomes. Les tests ont été produits en utilisant les différents paramètres de l'emploi de temps, croisement, et la probabilité de mutation. Croisements à deux points mis en œuvre au calendrier pour obtenir la solution optimale en utilisant diverses probabilités de croisement. Ce travail recommande d'améliorer la fonction de remise en forme et utiliser un mécanisme de sélection différent de l'algorithme.

Mots clés : Algorithme génétique, algorithme bio-inspiré, emploi de temps

Abstract

This dissertation offers the results of an investigation, and optimization of scheduling system in M'sila University (as a case study) using the genetic algorithm (GA). GA techniques are useful for solving real-world scheduling problem such as timetable which is a complex work and usually done manually.

This work focuses on scheduling courses timetable to allocate events (time, subject, and lecturer) in an appropriate way by using the available resource and assists to avoid conflicts. The algorithms explored different operator of GA such as crossover, mutation, and selection mechanism that's applied to set of chromosomes. The testing has been produced using different parameters of timetable, crossover, and mutation probability. Two point crossovers implemented to the timetable to obtain the optimal solution using various probabilities of crossover. This work recommended to enhance the fitness function and used different selection mechanism to the algorithm.

Keywords:genetic algorithm, bio-inspired, schedule problem.

ملخص

تقدم هذه الرسالة نتائج التحقيق ، تحسين نظام الجدولة في جامعة المسيلة (كدراسة الحالة) باستخدام الخوارزمية الجينية (AG). تقنيات AGمفيدة لحل مشكلة الجدولة في العالم الحقيقي مثلالجدول الزمني الذي هو عمل معقد وعادة مايتم ذلكيدويا.

يركز هذا العمل على جدولة مواعيد الدوراتلتخصيص الأحداث (الوقت والموضوع والمحاضر) فيالطريقة المناسبة باستخدام الموارد المتاحة وتساعد علىتجنب الصراعات. الخوارزميات استكشاف مختلف المشغل AG مثل التبادل، والتحول، وآلية الاختياريتم تطبيقه على مجموعة من الكروموسومات. كان الاختبار المنتج باستخدام معايير مختلفة للجدول الزمني، واحتمالية الطفرة. اثنين من عمليات الانتقالتنفذ للجدول الزمني للحصول على الحل الأمثلباستخدام احتمالات مختلفة من التبادل. هذه العمل أوصت لتعزيزوظيفة اللياقة واستخدام آلية اختيار مختلفة للخوارزمية

كلمات البحث :.خوارزمية الجينُية، الخوارزميات المستوحاة من الطبيعة، الجدول الزمني.

Liste des Acronymes

FA : Firefly Algorithm**.**

PSO: Particle Swarm Optimization.

CPSO: Cooperative Particle Swarm Optimization**.**

DBPSO : Discrete Binary Particle Swarm Optimization.

BPSO : Binary Particle Swarm Optimization.

APSO : Adaptive Particle Swarm Optimization.

CSP : programmation de satisfaction de contraintes.

ACPSO : Adaptive Cooperative Particle Swarm Optimization.

AG : Algorithme génétique.

Table des matières

Chapitre 3 : Résultats et discussion

Liste des figures

Introduction générale

Le problème de la planification d'horaire de travail est omniprésent dans beaucoup de domaines professionnels à savoir les usines ,où des pièces doivent être cheminées à travers nombreuse machines, la gestion de ce trafic doit répondre à certaines espérances, comme l'exploitation maximale des machines.

La planification du temps consiste à allouer des ressources données à des objets dans un intervalle de temps, pour satisfaire au mieux ensembles des finalités à l'instar de l'amélioration de la qualité de service et l'amélioration des sur constance de travail.

Parmi la vaste famille des problèmes de planification d´horaire, on trouve celui de la confection d´emploi du temps dans les établissements éducatifs, notamment dans les universités qui consomment de nombreuses ressources humaines et donc financières. Ce problème est très important. En effet un mauvais emploi du temps influe directement et négativement sur le niveau de l´acquisition des étudiants.

Le problème de l´emploi du temps est un problème ardu dont la réalisation à la main est une tâche draconienne qui peut mobiliser plusieurs personnes plusieurs jours de travail. Sans oublier, que toute modification des données du problème peut complètement remettre en cause la solution trouvée.

Ces difficultés, ont induit l'idée d'assister par ordinateur l'élaboration des emplois du temps en adoptant des outils basés sur des algorithmes d'optimisation robustes permettant de faciliter cette tâche. Ainsi depuis la fin des années cinquante, plusieurs contributions relatives au problème de l'emploi du temps sont apparus ;cela est dû, essentiellement, à sa complexité, sa grande taille et surtout au nombre important de problèmes pratiques que ce dernier regroupe.

D'une manière générale, le problème de l'emploi du temps consiste à définir un certain nombre d'affectations qui permettent d'assigner plusieurs ressources(humaines, matérielles,…etc.) sur une période de temps, tout en respectant les contraintes imposées par les entités citées (disponibilité des ressources humaines ,matérielles,…etc.).

Les problèmes de l'emploi du temps s'avèrent être NP difficile et la taille de leurs instances se caractérisent souvent par leur très grande taille.

Les contraintes considérées peuvent différer d'un problème à un autre suivant la spécificité ainsi que les caractéristiques attendues de l'emploi du temps recherché. Les contraintes sont souvent classées en deux catégories, la première regroupe les contraintes dures (un emploi du temps qui ne satisfait pas ce genre de contraintes est infaisable ou inacceptable), la seconde catégorie regroupe des contraintes (appelées souvent contraintes molles, souples ou de préférence) dont la satisfaction a différent degrés d'importances mais dont le non-respect n'empêche pas une application plus au moins acceptable de l'emploi du temps trouvé. Typiquement ces contraintes (de préférence) sont utilisées pour exprimer ce que doit être un« bon » emploi du temps. Ces contraintes sont plus difficiles à formaliser que les contraintes dures et leur traitement est plus délicat. Ainsi la majorité des approches existantes relaxent les contraintes de préférence et les introduisent comme une fonction objectif dont l'optimisation permet de se rapprocher le plus possible de la satisfaction des contraintes.

De ce fait, ce sont principalement les modèles mono objectifs qui sont utilisés lors de la résolution de ce type de problème. Notre étude s'est concentrée sur le fait que le problème de l'emploi du temps est lui aussi un problème d'optimisation.

Ce mémoire est organisé en trois chapitres. Dans le premier, nous introduisons les notions liées aux problèmes de planification d'horaires de travail et les différents types de plannings dans différents domaines de travail avec un aperçu des méthodes utilisées pour la réalisation de ces plannings.

Dans le deuxième chapitre, nous introduisons les algorithmes liées aux bio-inspiration. Nous donnons les différents algorithmes essaim de particule et les algorithmes évolutionnaires.

Un intérêt particulier sera porté sur l'étude des algorithmes génétiques.

Dans le dernier chapitre, nous proposons un algorithme génétique pour la résolution du problème d'emploi du temps.

Chapitre 1

Elaboration de planning

1.1. Introduction

Un emploi du temps est un calendrier de travail, où figure à la fois le temps, l'affectation des enseignants, des salles et des modules enseignés. La gestion des emplois du temps est une tâche très complexe, cette complexité vient du fait qu'on a un ensemble de contraintes à respecter, et des conflits à résoudre. Le problème de l´emploi du temps est un problème ardu dont la réalisation à la main est une tâche draconienne qui peut mobiliser plusieurs personnes plusieurs jours de travail. Sans oublier, que toute modification des données du problème peut complètement remettre en cause la solution trouvée. Pour cela nous définissons les différents types de plannings dans différents domaines de travail et plus particulièrement dans le domaine pédagogique.

1.2. La problématique de la planification d'horaires de travail

Les problèmes de planification d'horaires de travail se retrouvent autant dans les entreprises d'industrie que dans les services publics tels que : la santé, l'éducation etc…

La planification d'horaires de travail est un processus très complexe, qui vise à organiser des activités humaines (principalement de travail) dans le temps et à optimiser l'utilisation des ressources, de façon à couvrir un besoin exprimé par une charge de travail prévisionnelle sous diverses contraintes. Elle aboutit à des programmes définissant les horaires de travail et de repos de la force de travail [5].

Pour mieux cerner ce qui est la planification et la complexité à sa réalisation, ons'intéresse à un ensemble de questions :

1.2.1. Qu'est-ce que la planification ?

La planification est un instrument de gestion dont l'objectif est d'aboutir à des programmes permettant d'organiser et planifier le travail des salariés afin de rester pérenne dans l'économie globale. Ceci passe par la détermination des capacités de tout un chacun et par le recensement des activités futures et des besoins en personnel.

La planification vise à affecter les ressources humaines pour chaque intervalle de temps sur un horizon donné, de telle manière que les besoins par intervalle soient couverts et que les différentes contraintes soient satisfaites [5].

1.2.2. Qu'est-ce qu'un planning ?

Les plannings 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 [13]. En effet, ils apparaissent dans les cas suivants :

- Si le travail doit être assuré pendant plus d'une journée, il faut prévoir la succession de plusieurs personnes sur le même poste dans la journée. Un outil d'aide est nécessaire lorsque le nombre de postes dépasse la quinzaine, par exemple gérer les absences imprévues des salariés.
- Si le travail doit être assuré pendant plus de 35 heures par semaine, un outil automatique devient indispensable lorsque le nombre de postes dépasse la trentaine pour gérer la succession de plusieurs personnes dans la semaine, ainsi que les absences imprévues.

Les plannings peuvent être utilisés pour planifier les horaires de présences du personnel ou les tâches effectuées par le personnel :

- **Planning des horaires de présence** : ce type de planning est utilisé pour prévoir les horaires de présence du personnel sans préciser les tâches journalières à effectuer soit pour des raisons de sécurité, soit pour une meilleure souplesse.
- **Planning des tâches** : ce type de planning est utilisé dans les entreprises à haute technicité, comportant plusieurs métiers et compétences distincts, où il est souhaitable d'affecter le personnel en fonction des tâches. Ce qui exige une décomposition fine des opérations et le repérage des tâches que chaque personne est capable d'accomplir.

Les plannings peuvent être journaliers (spécifiant les pauses et périodes de travail de la journée de chaque employé), hebdomadaires (utilisés pour une paie hebdomadaire), mensuels (utilisés pour le calcul des coûts pour les besoins de la paiemensuelle) ou annuels (permettant de gérer les congés annuels des employés).

Selon leur spécificité et les branches d'activités concernées, les plannings portent différents noms. Un planning spécifiant les programmes de travail de chaque employé nominativement sur un horizon (un intervalle de temps où un planning est élaboré) d'un mois est appelé tableau de service. Lorsque le planning représente les programmes de travail et de repos non nominatifs sur un nombre entier de semaines, on par le de grille de travail.

Certains plannings sont cycliques, s'ils reflètent une certaine périodicité des horaires individuels c'est à dire si au bout d'une durée D (mesurée généralement en semaine), le salarié retrouve son planning de départ. Autrement, ils sont dits acycliquesc' est à dire ils sont différents chaque semaine.

1.2.3. A Quoi sert un planning ?

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

- aux entreprises exerçant une activité continue ou quasi-continue de répartir convenablement leur personnel (compagnies aériennes, entreprises de transports, hôpitaux, etc…),
- aux entreprises cherchant à se rendre plus accessibles à la clientèle d'étaler les horaires d'ouverture (grands magasins, banques, etc…),
- à toutes les entreprises de surmonter leur exigences de productivité et de mieux gérer les présences et absences de leur personnel. Les situations où un planning est utile son nombreuses. Elles justifient l'existence de différentes formes de plannings dans un

même système : plannings à court, moyen et à long terme [5].

1.2.4. Comment est évalué un planning ?

Pour que les plannings élaborés soient satisfaisants, ils doivent vérifier un ensemble de contraintes et établir un meilleur compromis entre les différents acteurs (exemple : le chef d'entreprise, le planificateur, le commercial, le syndicaliste et le salarié).

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 litigieux 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 plannings) rend d'autant plus difficile la négociation, car les opinions sont plus difficiles à formuler [19]. Les moyens informatiques apportent une aide certaine notamment dans l'acquisition et la confrontation des données individuelles.

1.2.5. Qui peut se charger de l'élaboration d'un planning ?

Dans la plupart des entreprises, cette tâche peut être centralisée ou déléguée à des cadres de l'entreprise appelés planificateurs. Le planificateur doit prendre la décision qui correspond le mieux aux préférences des différents acteurs, justifier son choix, car son expérience de la tâche fait de lui un interlocuteur privilégié pour évaluer rapidement et effectuer des jugements de l'orientation à donner à la recherche de solutions de meilleur qualité afin d'aboutir à un choix pertinent. Une collaboration réussie doit permettre au planificateur de participer efficacement à l'élaboration des plannings. La génération automatique des plannings joue un rôle primordial.

1.3. Différents types de plannings

Dans la construction de plannings d'horaires de travail, si créer un planning optimisé d'une journée est aisé, mais créer un bon planning pour un mois ou une année est beaucoup plus complexe. En plus de la complexité combinatoire du problème, il faut tenir compte de la diversité des contraintes applicables et qui sont souvent contradictoires.

1.3.1. Notions fondamentales

Parmi les objectifs de la théorie de complexité est de classer les problèmes en fonction des ressources (temps de calcul, espace mémoire, etc…) nécessaires à leur résolution algorithmique. Ceci a montré qu'il existe des problèmes qui ont une solution calculable mais dont toute réalisation effective sur une machine est pratiquement inutilisable parce que le temps de calcul ou la place mémoire nécessaire sont trop importants. L'enjeu est donc de discerner

entre les problèmes qui ont une solution réalisables et ceux qui, quelles que soient les améliorations futures des machines, ne peuvent intrinsèquement avoir une telle solution.

Quand on prend pour critère le temps d'exécution, on exprime cette frontière en considérant qu'un problème est réalisable si son temps d'exécution est polynomial en fonction de la taille de l'entrée. Si le degré du polynôme est élevé cette notion devient un peu irréalisable, mais la distinction entre temps polynomial et temps non polynomial donne naissance à une classification des problèmes qui est très utile pratiquement[27].

1.3.1.1. Notion de complexité

D'une manière générale, pour résoudre un problème, on est appelé à trouver l'algorithme le plus efficace. Cette notion d'efficacité induit normalement toutes les ressources de calcul nécessaires pour exécuter un algorithme. Or, le temps d'exécution est généralement le facteur dominant pour déterminé si un algorithme est assez efficace pour être utilisé dans la pratique, pour cela on se concentre principalement sur cette ressource.

On appelle complexité en temps d'un algorithme le nombre d'instructions élémentaires mises en œuvre dans cet algorithme afin de résoudre un problème donné.

Une instruction élémentaire sera une affectation, une comparaison, une opération algébrique, la lecture et l'écriture etc.… Mais comme le décompte précis de toutes les instructions d'un programme risque d'être assez pénible, et qu'entre deux exécutions du même algorithme avec un jeu de paramètres différent, le nombre d'instructions exécutées peut changer, on se contentera, en général, d'apprécier un ordre de grandeur de ce nombre d'instructions. C'est ce qu'on désigne sous le nom de complexité de l'algorithme. Donc pour mesurer la complexité temporelle d'un algorithme, on s'intéresse plutôt aux opérations les plus coûteuses[27] :

- Racine carrée, Log, Exp, Addition réelle…
- Comparaisons dans le cas des tris…

On dit que $f(n) = O(g(n))$($f(n)$ est de complexité$g(n)$), chaque fois qu'il existe k et $n_0$tels que :

$$n > n_0 \Rightarrow f(n) < \mathrm{kg}(n) \qquad (1.1)$$

1.3.1.2. Algorithmes polynomiaux et Algorithmes exponentiels

Un algorithme polynomial est un algorithme dont la complexité est O(p(n)) où p est une fonction polynomiale et n dénote la longueur de données. Tout algorithme dont la complexité ne peut être borné par un tel polynôme d'ordre n, est un algorithme exponentiel (bien que cette

définition inclue certaines complexités non-polynomiales comme n* log (n), qui n'est pas considéré comme fonction exponentielle)[27].

1.3.1.3. Problèmes combinatoires

Un problème combinatoire est un problème où il s'agit de trouver la meilleure combinaison possible de solutions. Un tel problème peut être soit un problème de décision, un problème de recherche ou un problème d'optimisation, selon la question à laquelle on est censé répondre[27].

1.3.1.4. Problème de décision

Un problème de décision est un problème où la résolution se limite à la réponse par « oui » ou par « non » à la question de savoir s'il existe une solution au problème Par conséquent, il n'est pas nécessaire de trouver la solution proprement dite[27].

1.3.1.5. Problème de recherche

Dans ce cas précis, la résolution du problème ne s'arrête plus au point de savoir si le problème admet ou non une solution. L'algorithme doit être en mesure de fournir la solution si celle-ci existe. Par conséquent, tout problème de décision peut être étendu à un problème de recherche[27].

1.3.1.6. Problème d'optimisation

Un problème d'optimisation est obtenu à partir d'un problème de recherche en associant à chaque solution une valeur. Il consiste à trouver parmi un ensemble de solutions potentielles celle qui répond le mieux à certains critères décrits sous forme d'une fonction objectif à maximiser ou minimiser c'est à dire on cherchera une solution de valeur optimale, minimale, si on minimise la fonction objectif, et maximale, si on la maximise[27].

1.3.1.7. La réduction polynomiale

On dit qu'un problème P1 se réduit polynomiale en un problème P2, s'il existe un algorithme polynomial A construisant, à partir d'une donnée D1 de P1, une donnée D2 de P2 telle que la réponse pour D1 soit oui si et seulement si la réponse pour D2 est oui[27].

1.3.1.8. La réduction de Turing

La réduction polynomiale permet de comparer les problèmes de décision. La réduction de Turing permet de comparer les problèmes de recherche. On dit qu'un problème de recherche P1 se réduit polynomiale à un problème de recherche P2 par la réduction de Turing s'il existe pour résoudre P1 un algorithme A1 utilisant comme sous–programme un algorithme A2

résolvant P2, de telle sorte que la complexité de A1 est polynomiale, quand on évalue chaque appel de A2 par une constante[27].

1.3.2. Les différentes classes de complexité

1.3.2.1. La classe P

La classe **P** contient tous les problèmes relativement faciles c'est à dire ceux pour lesquels on connaît des algorithmes efficaces. Plus formellement, ce sont les problèmes pour lesquels on peut construire une machine déterministe (e.g. une machine de Turing2) dont le temps d'exécution est de complexité polynomiale (le sigle **P** signifie « **P**olynomial time »)[27].

1.3.2.2. La classe NP

Les problèmes de la classe **NP** sont ceux pour lesquels on peut construire une machine de Turing non déterministe dont le temps d'exécution est de complexité polynomiale (le sigle NP provient de « Non déterministe Polynomial time ») (et non de « Non Polynomial).

Contrairement aux machines déterministes qui exécutent une séquence d'instructions bien déterminée, les machines non déterministes ont la remarquable capacité de toujours choisir la meilleure séquence d'instructions qui mène à la bonne réponse lorsque celle-ci existe. Ce concept abstrait est en fait la base de toute la théorie de la NP-complétude[27].

1.3.2.3. La classe NP-complets

Parmi l'ensemble des problèmes appartenant à NP, il en existe un sous ensemble qui contient les problèmes les plus difficiles : on les appelle les problèmes NP complets.

Un problème NP-complets possède la priorité que tout problème dans NP peut être transformé (réduit) en celui-ci en temps polynomial. C'est à dire qu'un problème est NP-complets quand tous les problèmes appartenant à NP lui sont réductibles. Si on trouve un algorithme polynomial pour un problème NP-complets, on trouve alors automatiquement une résolution polynomiale de tous les problèmes de la classe NP[27].

1.3.2.4. La classe NP-difficiles

Un problème est NP-difficile s'il est plus difficile qu'un problème NP-complet, c'est à dire s'il existe un problème NP-complet se réduisant à ce problème par la réduction de Turing.

Pour ce qui suit, on évoquera les différents types de plannings et les approches utilisées pour réaliser ces types de plannings[27].

1.3.3. Types de plannings dans le domaine de la santé

Les plannings dans les domaines de la santé sont des calendriers de travail où figurent à la fois le temps, et l'affectation des personnels (jours et horaires de travail, congés et repos). Ils sont établis au niveau de chaque équipe, ils sont à la fois une tâche, un document d'organisation du travail, et un élément contribuant à la gestion administrative du personnel. Cette tâche est parmi les plus difficiles et les plus délicates. Difficile parce qu'elle repose sur la recherche de solutions combinatoire, répond à des contraintes multiples

1.3.4. Types de plannings dans le domaine de transport

Le transport est une activité complexe qui fait intervenir des investissements lourds, du personnel qualifié et une informatique très coûteuse, En effet, dans le transport routier, il est toujours nécessaire de gérer aux mieux les ressources existantes en optimisant les investissements

1.3.5. Types de plannings dans le domaine de la pédagogie

La confection d'horaires (ou confection d'emploi du temps) dans les établissements scolaires est un travail très important, difficile à réaliser, c'est typiquement un problème de résolution de contraintes, NP-complet, dont la solution n'est pas, a priori connue dans le cas général. Pour fournir une solution , nécessite d'être capable de s'adapter aux changements dynamiques de l'environnement en tenant compte de la diversité des contraintes telles que l'interdépendance des programmes d'enseignement, la multitude des matières étudiées et les contraintes sur ces matières(cours, cours magistraux, TD, TP…), la durée des cours, les contraintes de disponibilité des enseignants, la disponibilité limitée des salles. C'est un problème qui peut être défini comme un problème qui fait assigner quelques évènements dans un nombre limité de périodes. Il peut être divisé en deux catégories principales : la confection d'horaires des cours et la confection d'horaires des examens. Ces problèmes sont soumis à beaucoup de contraintes qui sont d'habitude divisées en deux catégories : « les contraintes dures » et« les contraintes souples »[3].

La confection de plannings d'horaires est donc une tâche très difficile et sa solution manuelle peut exiger beaucoup d'effort ce qui a attiré énormément l'attention de la communauté scientifique. Comme notre travail se rapporte au problème de résolution d'emploi du temps d'université, on va essayer de voir l'historique des différentes recherches étudiées dans la littérature :

Une large variété d'approches et modèles ont été proposés pour traiter une variété de problèmes d'emploi du temps. Les problèmes s'étendent de la construction des emplois du temps semestriels ou annuels dans les universités, écoles ou collèges aux emplois du temps d'examens à la fin de ces périodes. 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 à changer 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. Par conséquent le besoin de la génération automatisée d'emploi du temps augmente et ainsi le développement d'un système de génération d'emploi du temps qui produit des solutions valables est essentiel. En conséquence, pendant les 30 dernières années, beaucoup d'approches liées à l'automatisation des emplois du temps ont été publiées aux conférences et journaux.

De plus, plusieurs applications ont été développés et mises en œuvres avec divers succès [25]. Les premières techniques employées dans la résolution du problème d'emploi du temps ont étaient basées sur la simulation de l'approche humaine dans la résolution du problème, ces techniques ont été appelées « les heuristiques directes »,elles sont basées sur l'idée de créer un emploi du temps partiel en planifiant d'abord le cours le plus contraint, ensuite, cette solution partielle est étendue jusqu'à ce que tous les cours seront planifiés. L'étape suivante été l'application des techniques générales telles que la programmation linéaire et la coloration de graphes pour résoudre ce problème d'emploi du temps. De là, les premières publications sur la construction d'emploi du temps employant ces techniques générales sont attribuées à Kuhn et Haynes [22].

L'intérêt de génération d'emploi du temps a augmenté dramatiquement dans les années 60 principalement en la raison de la disponibilité d'ordinateurs pour exécuter les algorithmes développés. En 1994, Corne, a fait une enquête sur l'application des algorithmes génétiques au problème d'emploi du temps et a discuté les futures perspectives de telles approches en comparant les résultats obtenus avec ceux obtenus avec d'autres approches[22].

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 [25].

1.4. Les différentes approches de résolution

1.4.1. Les méthodes séquentielles

Ces méthodes ordonnent les évènements en les assignant séquentiellement dans des périodes de temps valides pour qu'aucun évènement dans une période ne soit en conflit avec un autre dans une période de temps donné. Dans les méthodes séquentielles, les problèmes de confection d'horaires sont généralement représentés par des graphes où les évènements (cours/examens) sont représentés par des nœuds et les conflits entre les évènements sont représentés par les arcs[30]. Par exemple si quelques étudiants doivent suivront deux évènements, il y a un arc entre les nœuds qui représentent le conflit. La construction d'un emploi du temps peut donc être modelée comme un problème de coloration de graphe.

Chaque fois la période dans l'emploi du temps correspond à une couleur et les nœuds du graphique sont colorés de telle façon qu'aucun des nœuds adjacents ne soit coloré par la même couleur. Une variété d'heuristiques de coloration de graphe pour la construction des conflits d'emploi du temps est disponible dans la littérature. Ces heuristiques ordonnent les évènements basés sur une évaluation de comment il est difficile de les prévoir.

Les heuristiques qui sont souvent employées sont

- le plus grand degré d'abord :Les évènements qui ont un grand nombre de conflits avec d'autres évènements sont prévus tôt. Le raisonnement est que les évènements avec un grand nombre de conflits sont plus difficiles à prévoir et donc doivent être abordé d'abord.
- le plus grand degré pondéré :C'est une modification du plus grand degré d'abord où le poids de chaque conflit est représenté par le nombre d'étudiants impliqués dans le conflit.
- le degré de saturation :Dans chaque pas de la construction de l'emploi du temps, l'événement qui a un nombre petit de périodes valables disponibles pour la planification dans l'emploi du temps est choisi lointainement.

1.4.2. Les méthodes basées contraintes

Dans ces méthodes, le problème d'emploi du temps est modelé comme un jeu de variables (c.-à-d. les évènements), les valeurs (c.-à-d. les ressources comme les pièces et les périodes) vont être assignées pour satisfaire un certain nombre de contraintes .D'habitude quelques règles sont définies pour l'assignation de ressources aux évènements. Quand aucune règle n'est

applicable à la solution partielle actuelle un retour arrière est exécutée jusqu'à une solution est trouvée qui satisfait toutes les contraintes[27].

1.4.3. Les méthodes méta heuristiques

Pendant les deux dernières décennies une variété d'approches méta euristiques comme le recuit simulé, la recherche tabou, les algorithmes génétiques, les colonies de fourmis et les approches hybrides ont été étudiées pour la résolution du problème d'emploi du temps. Les méta heuristiques commencent par une ou plusieurs solutions initiales et emploient les stratégies de recherche qui essayent d'éviter des optimums locaux. Tous ces algorithmes de recherche peuvent produiront des solutions de qualité mais ont souvent un coût informatique considérable.

Les algorithmes génétiques sont une classe des algorithmes de recherche stochastiques qui emploient la génétique comme modèle pour la résolution du problème. L'application de la reproduction, la sélection et la mutation peut donner beaucoup d'avantages pour la survie de nouvelles générations. Cet algorithme est très indiqué pour les problèmes distribués par nature et les problèmes susceptibles d'évolution dynamique. Cette coopération peut prendre la simple forme d'un passage de relais entre la méta-heuristique et la technique locale, comme elles peuvent être entremêlées de manière plus complexe .Parmi les modèles proposés pour confectionner des emplois du temps citons quelques exemples :

Les méthodes évolutives [1] : où l'approche proposée est fondée sur une programmation par contraintes utilisant un algorithme génétique comme moteur d'optimisation.

1.5. Conclusion

Dans ce chapitre, nous avons présenté le problème des emplois du temps des cours est un processus complexe, c'est un problème d'optimisation combinatoire très difficile à résoudre. Car une solution au problème est représentable par un ensemble de propriétés. Le but est d'atteindre la meilleure combinaison de ces propriétés. Une autre cause de complexité du problème est le volume du problème .Parmi tous les types de plannings cités, c'est sur les plannings pédagogiques que nous allons porter notre intérêt, et plus particulièrement sur les plannings ou emploi du temps des cours d'université.

Dans le chapitre suivant nous présentons les systèmes bio-inspires et leurs algorithmes.

Chapitre 2

Les algorithmes bio-inspiré concepts et méthodes

2.1. Introduction

Récemment, les algorithmes bio-inspire dans des environnements combinatoire a suscité un intérêt croissant, en raison de son utilité pratique. De nombreuses applications peuvent s'exprimer sous la forme de problème d'optimisation combinatoire. Le cas le plus fréquent est celui ou la fonction objectif varie au fil du temps :des exemples typiques sont le routage dynamique de véhicules ,la planification d'emploi de temps, la planification de stocks, le suivi d'un objet mobile, …etc.

Il arrive aussi que la fonction objective soit incertaine ou bruitée ,en raison d'erreurs de simulation, de mesure ou d'approximation. En outre, les variables de conception ou les conditions environnementales peuvent également être perturbées, où changer au fil du temps.

Pour ces problèmes d'optimisation d'emploi de temps, l'objectif d'une Méta-heuristique efficace est de trouver la solution globale et de la suivre en permanence dans les différents environnements, ou encore de trouver une solution robuste, qui fonctionne de manière optimale, en présence d'incertitudes.

2.2. Bio-inspiration

La bio-inspiration est un changement de paradigme qui conduit des concepteurs à s'inspirer de la nature pour développer de nouveaux systèmes. La bio inspiration s'appuie souvent sur le biomimétisme .Comme lui, elle peut puiser son inspiration tant le monde des végétaux que des animaux et champignons, ou des bactéries et virus .Elle a déjà contribué à des applications dans des domaines aussi variés que l'aéronautique, la pharmacie, la marine, la médecine, la chimie verte, les matériaux composites, la robotique, l'intelligence artificielle ou encore les nanotechnologies.

Elle est utilisée par des acteurs non industriels(dans des domaines tels que le design, l'art, l'architecture, l'urbanisme ,l'enseignement, la conception de logiciel, d'algorithme génétique et/ou d'algorithme évolutionniste) ...et de manière mieux connue du grand public par les ingénieurs qui parfois lui adaptent des méthodes issues de la rétrogénierie [32].

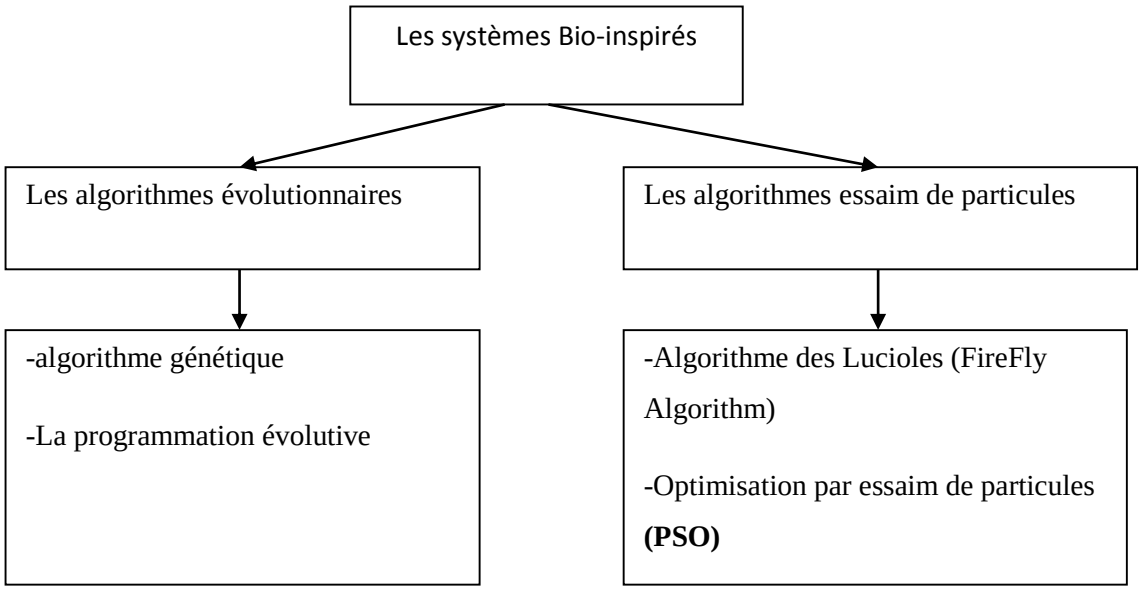


Figure2.1 : Classification des systèmes bio-inspirés

2.2.1. Les algorithmes essaim de particules

Les algorithmes essaim de particules sont de plus en plus développées et proposées pour la résolution de problèmes complexes, parmi ces algorithmes : l'algorithme de chauves-souris (Bat 49 Algorithm – BA), l'algorithme de lucioles (Firefly Algorithm – FA), l'algorithme de la pollenisation des fleurs (Flower Pollination Algorithm – FPA), l'optimisation inspirée pigeon (Pigeon Inspired Optimization), … etc.

Dans cette section, nous allons aborder les deux algorithmes de swarm intelligence les plus populaires et les plus utilisés à ce jour, qui sont : l'optimisation par essaims particulaires (Particle Swarm Optimization – PSO) et les Algorithme des Lucioles (FireFly Algorithm)

2.2.1.1. Algorithme des Lucioles (FireFly Algorithm)

a) Inspiration

Les lucioles (en anglais FireFly) sont de petits coléoptères ailés capables de produire une lumière clignotante froide pour une attraction mutuelle.

Dans le langage courant entre les lucioles, ils sont également utilisés synonymes bogues d'éclairage ou des vers luisants. Ce sont deux coléoptères qui peuvent émettre de la lumière, mais les lucioles sont reconnues comme des espèces qui ont la capacité de voler.

Ces insectes sont capables de produire de la lumière à l'intérieur de leur corps grâce à des organes spéciaux situés très près de la surface de la peau. Cette production de lumière est due à un type de réaction chimique appelée bioluminescence [35].

Les femelles peuvent imiter les signaux lumineux des autres espèces afin d'attirer des mâles qu'elles les capturent et les dévorent. Les lucioles ont un mécanisme de type condensateur, qui se décharge lentement jusqu'à ce que certain seuil est atteint, ils libèrent l'énergie sous forme de lumière. Le phénomène se répète de façon cyclique. L'Algorithme des lucioles développé par Yang [35] est inspiré par l'atténuation de la lumière sur la distance et l'attraction mutuelle mais il considère toutes les lucioles comme unisexes.

Figure 2.2 : Les Lucioles

b) Principe de fonctionnement de l'algorithme des Lucioles

L'algorithme Firefly [36] [35] [37] [38] a été introduit par XS Yang en 2008. Sa source d'inspiration est basée sur l'émission de la lumière, absorption de la lumière et le comportement attractif mutuelle entre les lucioles. Initialement, il a été développé pour résoudre les problèmes d'optimisation, mais plus tard il a été utilisé pour résoudre discrètement des problèmes tels que les vendeurs itinérants …etc.

En outre, il a également été utilisé dans le domaine du traitement d'image numérique, de compression et de clustering. L'algorithme tire son inspiration de comportement clignotant des lucioles. Les trois règles sont idéalisées en décrivant le FA algorithme [23][20][17]:

- Toutes les lucioles sont unisexes de telles sortes que seront attirées vers d'autres indépendamment de leurs sexes.
- L'attractivité des lucioles est directement proportionnelle à la luminosité entre eux, et est diminué une fois la distance entre deux lucioles augmente. Ainsi, pour tous les deux lucioles clignotantes quelconques, le moins lumineux sera attiré et se déplace donc vers le plus lumineux.
- La luminosité de la lumière clignotante peut être considérée comme une fonction objective qui devra être optimisé.
- La plupart des méthodes d'optimisation méta-heuristiques sont basé sur la génération aléatoire de la population initiale de solutions candidate possible. Pour cela le processus de l'algorithme de luciole commence avec l'initialisation de la population des lucioles et donc chaque luciole dans une population représente une solution candidate.
- La taille de la population détermine le nombre de solutions ou la taille de l'espace de recherche dont le but est d'orienter la recherche à la meilleure localisation.
- Dans l'étape suivante, chaque luciole est évaluée en fonction de leur condition physique (intensité lumineuse). À chaque nouvelle étape itérative, la luminosité et l'attirance de chaque luciole est calculée.
- La distance entre toutes lucioles peut être définie comme une distance cartésienne.
- La fonction de la distance développée est utilisée pour trouver la distance entre deux lucioles.
- La fonction de l'attractivité est définie en utilisant l'intensité lumineuse, la distance entre lucioles, et un coefficient d'absorption.
- Le mouvement de luciole est défini par une fonction de mouvement, en utilisant la position actuelle, l'attractivité et une marche aléatoire, après avoir comparé la luminosité de chaque luciole avec toutes les autres lucioles, les positions des lucioles sont mises à jour en se basant sur les règles de connaissances sur les lucioles et sur leurs voisins.
- Après le déplacement, la nouvelle luciole est évaluée et l'intensité de sa lumière est mise à jour. Pendant la boucle de comparaison en paire, la meilleure solution actuelle est la mise à jour d'une manière itérative. Le Processus de comparaison par pair est

répété jusqu'à la satisfaction des critères de résiliation.

- Les principales étapes de l'algorithme FireFly sont données Ci-dessous :

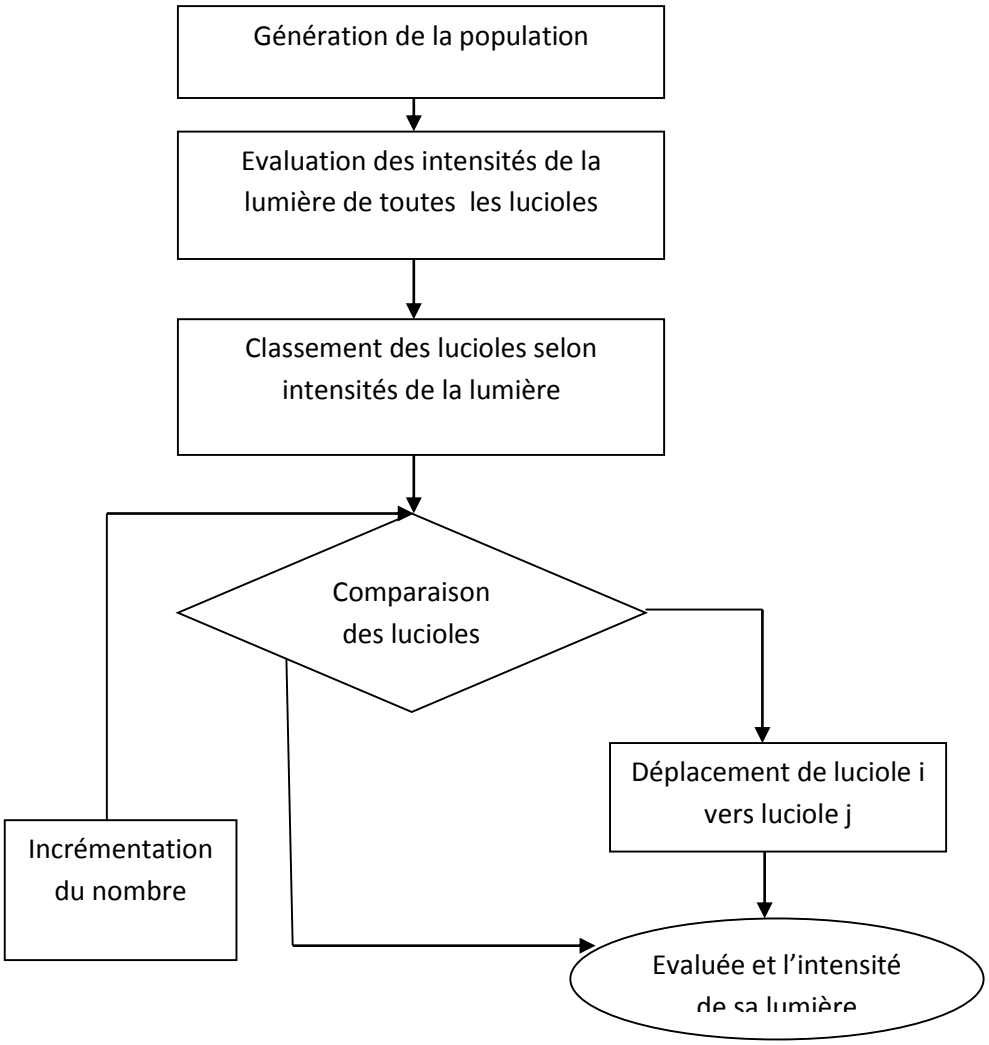


Figure 2.3 : Organigramme de fonctionnement d'un algorithme FireFly

Ou *I(i)*: est l'intensité de la lumière d'ième luciole.

c) Génération de la population initiale

Dans cette étape, l'algorithme des lucioles "Firefly algorithm" génère une population initiale qui représente un ensemble de solutions possibles.

Cette population initiale est généralement générée aléatoirement, le choix de la population initiale est très important car il peut rendre plus ou moins rapide la convergence vers l'optimum global.

Dans le cas où l'on ne connaît rien sur le problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche.

En contrepartie, le choix d'une population trop élevée peut augmenter considérablement le temps de calcul, et si la taille de la population est trop petite, il y aura une convergence prématurée car l'algorithme n'a pas un grand échantillon de l'espace de recherche.

d) Fonction d'évaluation

La fonction d'évaluation (fitness) en terminologie anglo-saxome [24] ou de coût, attribut à chaque luciole une valeur numérique qui représente un coût de performance, elle est utilisée pour coder la luminosité des Firefly. Grace à cette fonction l'algorithme converge vers l'optimum.

L'efficacité de l'algorithme en termes de pertinence de la solution et le temps de calcul dépend principalement de la fonction objective, pour cela elle doit définir les fonctions objectives de façon plus fidèle que possible.

Il existe deux types de fonction d'évaluation, soit mono critère ou multicritère :

- **Une fonction d'évaluation mono critère** : Signifie que la fonction dépend d'une seule et même fonction objectif. La résolution de la fonction d'adaptation (fitness), dans ce cas est simple et ne pose généralement aucun problème.
- **Une fonction d'évaluation multicritère** : Généralement, les problèmes d'optimisation doivent souvent satisfaire des objectifs multiples. Une méthode classique consiste à définir des fonctions objectifs élémentaires dont certains sont concurrents, traduisant chaque objectif à atteindre, et de les fusionner au sein d'une seule fonction.

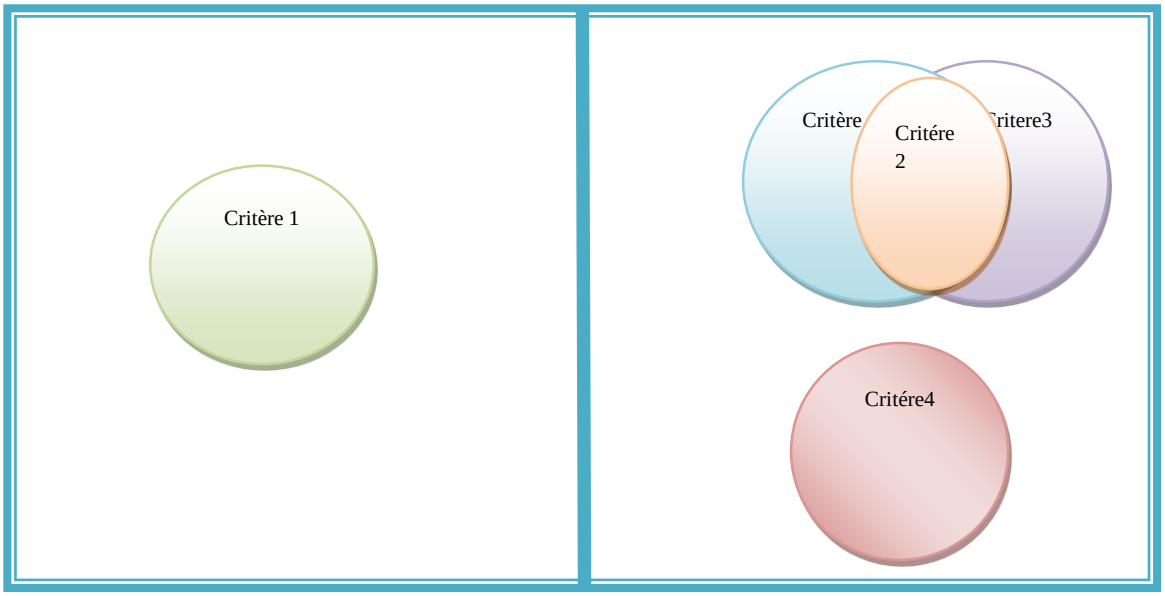


Figure 2.4 : Fonctions mono critère et multi critère

e) Classement

Le classement s'effectue par rapport à la fonction objective, donc dans notre algorithme le classement se fait selon l'intensité de la lumière de chaque luciole.Généralement le classement sert à déterminer le meilleur individu ou le mauvais individu, si la fonction objective cherche à maximiser les critères, le classement par ordre croissant et donc le meilleur est le maximum et le mauvais est le minimum, sinon si la fonction objective cherche à minimiser les critères, le classement se fait par ordre décroissant et donc le meilleur est le minimum et le mauvais est le maximum.

f) Déplacement et mise à jour

À chaque nouvelle étape itérative, la luminosité et l'attirance de chaque firefly est calculée, c'est la fonction objectif.

Après avoir comparé la luminosité de chaque luciole avec toutes les autres lucioles, les positions des lucioles sont mises à jour en se basant sur des règles de connaissance de la luciole et de leurs voisins, ces règles sont généralement la position initiale, la distance entre deux lucioles comparés et un mouvement aléatoire.

Après le déplacement, la nouvelle luciole est évaluée, sa position et son intensité de lumière sont mises à jour.

Pendant la boucle de comparaison en deux à deux, la meilleure solution est mise à jour de manière itérative. Le processus de comparaison par pair est répété jusqu'à la satisfaction des critères de résiliation [20].

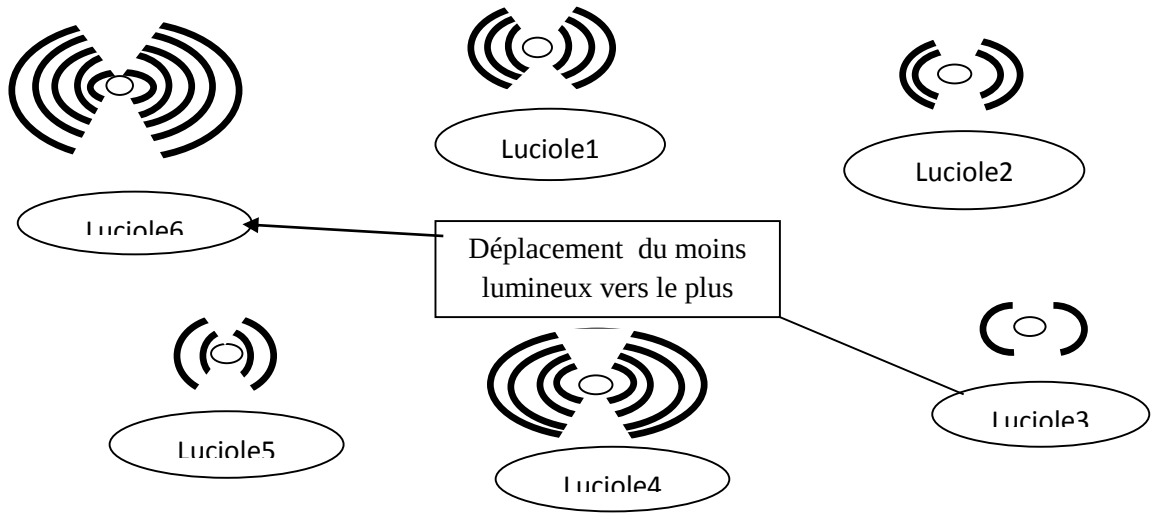


Figure 2.5 : Déplacement des lucioles dans une itération

g) Critère d'arrêt

Les étapes précédentes (déplacement et mise à jour) appliquées d'une manière itérative, cette boucle s'arrêt jusqu'à ce que la condition d'arrêt soit satisfaite. Cette condition correspond soit à un nombre maximum de génération fixée au départ, ou quand une solution satisfaisable proche de la solution optimale est atteinte. Aussi on peut arrêter la boucle quand les résultats de l'algorithme sont devenus stable, pour éviter la perte de temps.

Au cours de son évaluation, la population tend à converger, c-à-d, les lucioles tendent à se ressembler de plus en plus. Quand la population s'est uniformisée en grande partie, les lucioles fournissent une bonne approximation d'un optimum du problème. Si cet optimum n'est pas toujours l'optimum global, c'est généralement un optimum local proche (en qualité) de celui-ci.

2.2.1.2. Optimisation par essaim de particules (PSO)

L'optimisation par les essaims particulaires est un modèle stochastique, une technique d'optimisation basée population qui peut être appliquée à un large éventail de problèmes.

a) Principe général

L'optimisation par essaim de particules (en anglais, PSO – Particle Swarm Optimization) qui a été introduite par Kennedy et Eberhart [14] en 1995 est l'un des paradigmes les plus importants de la Swarm Intelligence. Il utilise un mécanisme simple qui imite les comportements d'essaims comme les volées d'oiseaux et les bancs de poissons pour guider les particules à la recherche de solutions globales optimales. Comme PSO est simple à mettre en œuvre, il a beaucoup progressé ces dernières années dans la résolution de problèmes d'optimisation du monde réel [40].

Le PSO s'inspire du comportement social basé sur l'analyse de l'environnement et du voisinage qui constitue une méthode de recherche d'optimum par l'observation des tendances des individus voisins. Chaque individu cherche à optimiser ses chances en suivant une tendance qu'il modère par son propre vécu. L'ensemble d'individus originellement disposés de façon aléatoire et homogène, que nous appellerons des particules, se déplacent dans l'espace de recherche et constituent, chacune, une solution potentielle.

Chaque particule dispose d'une mémoire concernant sa meilleure solution visitée ainsi que la capacité de communiquer avec les particules voisines (voir Figure 2.6). A partir de ces informations, la particule va suivre une tendance basée, d'une part, sur sa volonté à retourner vers sa solution optimale, et, d'autre part, sur son mimétisme par rapport aux solutions trouvées chez les particules voisines.

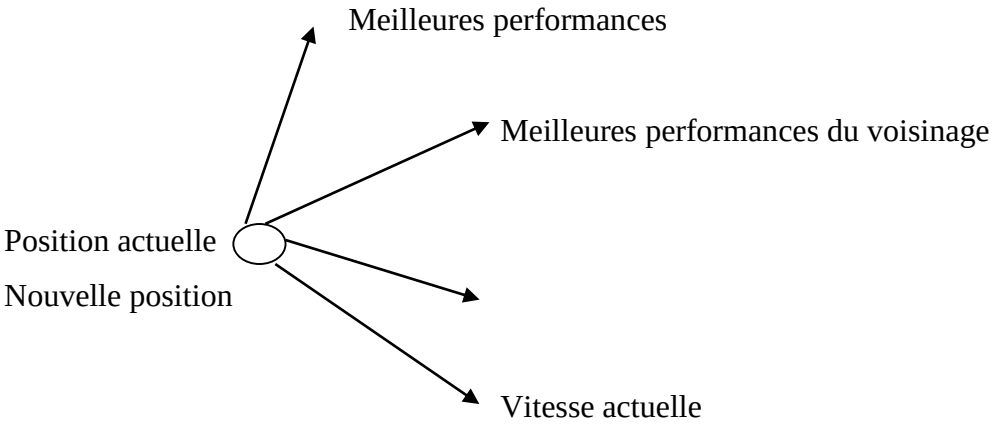


Figure 2.6 : Eléments du comportement des particules d'un essaim.

b) Formalisation

Un essaim de particules est caractérisé par :

- Nombre de particules de l'essaim ***nb***
- Inertie $\boldsymbol{\Psi}$
- Vitesse6 maximale d'une particule $\boldsymbol{Vmax}$
- Les coefficients de confiance $\boldsymbol{p1}$ et $\boldsymbol{p2}$ qui pondèrent le comportement conservateur(c'est-à-dire la tendance à retourner vers la meilleure solution visitée) et le panurgisme(tendance à suivre le voisinage).
- Topologie et taille du voisinage d'une particule qui définissent son réseau social L'inertie permet de définir la capacité d'exploration de chaque particule en vue d'améliorer la convergence de la méthode. Si l'inertie est de forte valeur, ceci implique une exploration globale. Au contraire, si l'inertie est de faible valeur, ceci indique une exploration locale.

Fixer ce facteur, revient à trouver un compromis entre l'exploration locale et l'exploration globale [4].Le calcul de la vitesse est alors défini par :

$$vit = \Psi.\ vit - 1 + p1.\ (\ xpbesti - xi(t)) + p2.\ (\ xvbesti - xi(t)) \qquad (2.1)$$

Les variables de confiance pondèrent les tendances de la particule à vouloir suivre son instinct ou son panurgisme. Les variables aléatoires $p1$ et $p2$ peuvent être définis de la façon suivante :$p1 = r1 \times c1$ et $p2 = r2 \times c2$

Où $r1$ et $r2$ suivent une loi uniforme sur 0. .1, $c1$ et $c2$ sont des constantes positives déterminées de façon empirique et suivant la relation $c1 + c2 \leq 4$

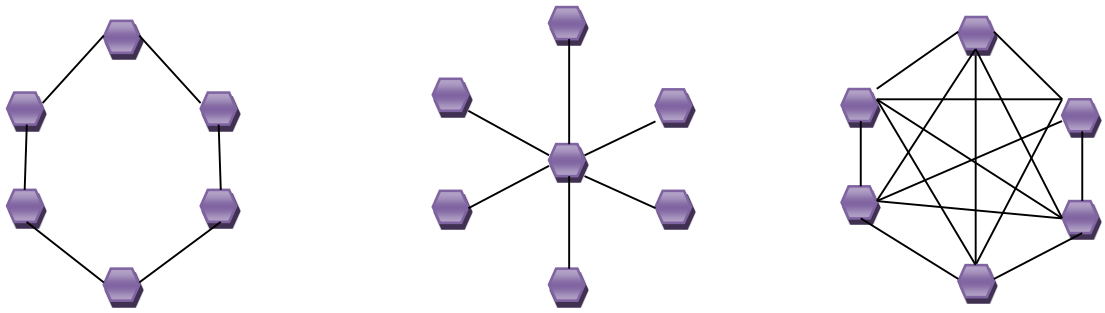
Une particule est caractérisée, à l'instant t par :

- Sa position dans l'espace de recherche $\boldsymbol{xi}$

- Sa vitesse $\boldsymbol{vi}$ ($\boldsymbol{t}$)
- Sa position de la meilleure solution par laquelle elle est passée $\boldsymbol{xpbesti}$
- La position de la meilleure solution connue de son voisinage $\boldsymbol{xvbesti}$
- Valeur de fitness de sa meilleure solution $\boldsymbol{pbesti}$
- Valeur de fitness de la meilleure solution connu du voisinage $\boldsymbol{vbesti}$

La topologie du voisinage définit avec qui chacune des particules va pouvoir communiquer. Il existe plusieurs topologies de voisinage dont les plus utilisées sont en anneau, en rayon et en étoile (voir Figure 2.7).

Dans la version originale de PSO, c'est une topologie entièrement connectée (i.e. étoile où chaque particule est reliée à toutes les autres) qui est utilisée. Cette version de PSO est appelée version globale (*Gbest*), car la particule est informée par la totalité des autres. Cette version a l'inconvénient majeur de ne pas donner lieu à une exploration suffisante, ce qui peut conduire à une stagnation dans un optimum local et donc à une convergence prématurée [7].



Anneau **Rayon** **Etoile**

Figure 2.7 : Les différentes topologies de voisinages.

De nombreuses variantes de la version originale, dites versions locales (*Lbest*), ont été proposées dans la littérature de PSO, afin d'améliorer sa convergence. Parmi ces variantes, nous pouvons citer la topologie en anneau qui utilise un graphe d'information statique (cette version est connue comme étant la plus utilisée) et la topologie en rayon où les particules ne communiquent qu'avec une seule particule centrale [4].

c) Les variantes de PSO

Au cours de ces dernières années, différentes améliorations ont été apporté à l'algorithme d'optimisation par essaims de particules traditionnel dans le but est de l'adapter à des problèmes spécifiques et d'améliorer ses performances. De ces améliorations, plusieurs variantes de PSO ont vu le jour pour résoudre des problèmes spécifiques comme le PSO coopératif (Cooperative Particle Swarm Optimization – CPSO) proposé par Van den Bergh et Engelbrecht [29]. Cette approche coopérative introduite améliore les performances au fur et à mesure que la dimension

du problème augmente comparé à l'algorithme conventionnel PSO qui a des difficultés à gérer une grande dimensionnalité de l'espace de recherche. Afin de traiter des problèmes binaires, Kennedy et Eberhart [13] ont proposé une version discrète et binaire de PSO (Discret Binary Particle Swarm Optimization – DBPSO). Dans ce modèle, une particule décidera « oui » ou « non », ou « vrai » ou « faux », « inclure » ou « ne pas inclure », etc. Cette variante a été reprise sous le nom de PSO binaire (Binary PSO – BPSO) [6, 12].

Une version modifiée de l'algorithme conventionnel PSO proposé par Guangyon [11] où le degré d'agrégation des particules de l'essaim a été introduit. La diversité des particules a été améliorée grâce au contrôle périodique de ce degré d'agrégation. Un algorithme PSO adaptatif (Adaptive Particle Swarm Optimization – APSO) qui dispose d'une meilleure efficacité de recherche que l'algorithme PSO conventionnel a été présenté par Zhan et al. [40]. PSO avec un poids d'inertie adaptatif est proposé par Wang [30] sous le nom d'algorithme PSO coopératif et adaptatif (Adaptive Cooperative Particle Swarm Optimization – ACPSO). L'apprentissage coopératif est obtenu en divisant un essaim d'Anneau Rayon Etoile grande dimension en plusieurs sous-essaims de petite dimension. Le poids de l'inertie adaptatif est utilisé pour contrôler l'équilibre de l'exploration et l'exploitation dans tous les sous-essaims de petite dimension.

Il y a plusieurs autres variantes de l'algorithme PSO dans la littérature que nous n'avons pas pu toutes énumérer en raison de leur grand nombre, plusieurs Survey récents [8, 16, 21] ont étudié en profondeur l'algorithme PSO, ses variantes et son application dans divers domaines.

2.2.2. Les algorithmes évolutionnaires

Les algorithmes évolutionnaires sont des techniques d'optimisation itérative, inspirées donc de la théorie de l'évolution de Darwin. Un algorithme évolutionnaire simule un processus d'évolution sur une population d'individus, dans le but de faire évoluer vers les optimums globaux du problème d'optimisation considéré.

Un algorithme évolutionnaire typique réunit trois composants :

1. Une population constituée de plusieurs individus représentant des solutions potentielles du problème posé (en optimisation combinatoire classique : configuration du problème).
2. Une fonction d'adaptation (fitness) qui évalue la performance d'un individu par rapport au milieu (en optimisation combinatoire : fonction du coût).
3. Un mécanisme d'évolution de la population composé de plusieurs opérateurs de modification et de sélection permettant (grâce à ces opérateurs prédéfinis), d'éliminer

certains individus et d'en créer de nouveaux. Le cycle d'évolution d'un algorithme évolutionnaire typique débute avec une population initiale (généralement obtenue de façon aléatoire) puis répète la boucle suivante (qui présente ce cycle d'évolution) :

a) mesurer l'adaptation (la qualité) de chaque individu de la population par le mécanisme d'évaluation,

b) sélectionner une partie des individus,

c) produire de nouveaux individus par recombinaisons des individus sélectionnés.

Ce processus se termine quand la condition d'arrêt est vérifiée, par exemple quand un nombre maximum de cycles ou un nombre maximum d'évaluations est atteint.

Les différences principales entre les algorithmes évolutionnaires, et les méthodes classiques d'optimisation combinatoire sont les phases de sélection et d'évolution. La sélection permet de choisir les meilleurs individus et c'est à partir d'eux que l'on construit la génération suivante. L'évolution quant à elle repose sur deux principaux opérateurs, la recombinaison qui combine plusieurs individus parents pour créer des individus enfants pour la génération suivante et la mutation qui altère légèrement certains individus.

D'une manière générale, on peut distinguer trois grandes classes d'algorithmes évolutionnaires :les algorithmes génétiques, la programmation évolutive et les stratégies d'évolution. Ces méthodes se différencient par leur manière de représenter l'information et par leur façon de faire évoluer la population d'une génération à l'autre.

2.2.2.1. La programmation évolutive

La programmation évolutive a été initialement introduite pour simuler l'intelligence qui est définie sur l'hypothèse suivante : la caractéristique principale de l'intelligence est la capacité d'adaptation comportementale d'un organisme à son environnement. Aujourd'hui la programmation évolutive s'est adaptée à l'optimisation combinatoire.

La programmation évolutive s'appuie sur un codage approprié du problème à résoudre et sur les opérations de mutation adaptées au codage. Le codage d'un tel algorithme dépend du problème à résoudre. Par exemple, pour un problème d'optimisation dans le domaine des réels, les individus d'une population seraient des vecteurs de réels.

Un cycle d'évolution typique pour la programmation évolutive est le suivant : chaque configuration de la population courante est copiée dans une nouvelle population. Les configurations sont ensuite mutées, conduisant à de nouvelles configurations.

L'ensemble des configurations entre ensuite dans une étape de compétition pour survivre dans la génération suivante.

2.2.2.2. L'algorithme génétique

L'algorithme génétique (AG) est un algorithme de recherche basé sur les mécanismes de sélection naturelle et de la génétique. Il combine une stratégie de "survie des plus forts" avec un échange d'information aléatoire mais structuré. Pour un problème pour lequel une solution est inconnue, un ensemble de solutions possibles est créé aléatoirement.

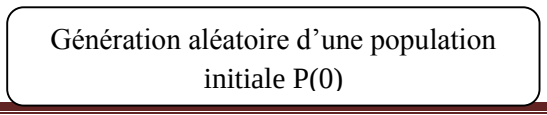
On appelle cet ensemble la population. Les caractéristiques (ou variables à déterminer) sont alors utilisées dans des séquences de gènes qui seront combinées avec d'autres gènes pour former des chromosomes et par après des individus. Chaque solution est associée à un individu, et cet individu est évalué et classifié selon sa ressemblance avec la meilleure, mais encore inconnue, solution au problème. Il peut être démontré qu'en utilisant un processus à une solution.

Comme dans les systèmes biologiques soumis à des contraintes, les meilleurs individus de la population sont ceux qui ont une meilleure chance de se reproduire et de transmettre une partie de leur héritage génétique à la prochaine génération. Une nouvelle population, ou génération, est alors créée en combinant les gènes des parents. On s'attend à ce que certains individus de la nouvelle génération possèdent les meilleures caractéristiques de leurs deux parents, et donc qu'ils seront meilleurs et seront une meilleure solution au problème. Le nouveau groupe (la nouvelle génération) est alors soumis aux mêmes critères de sélection, et par après génère ses propres rejetons. Ce processus est répété plusieurs fois, jusqu'à ce que tous les individus possèdent le même héritage génétique. Les membres de cette dernière génération, qui sont habituellement très différents de leurs ancêtres, possèdent de l'information génétique qui correspond à la meilleure solution au problème.

L'algorithme génétique de base comporte trois opérations simples qui ne sont pas plus compliquées que des opérations algébriques :

- Sélection
- Reproduction
- Mutation

L'algorithme génétique fut développé par Holland [9].



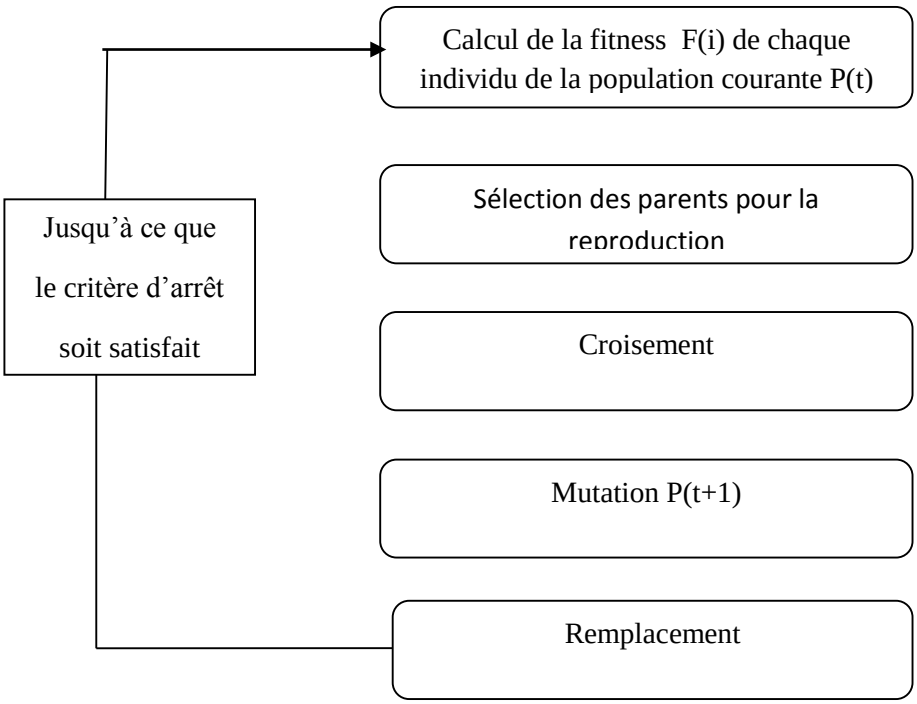


Figure2.8 Les opérateurs génétiques

a) L'opérateur d'initialisation

Cet opérateur est utilisé pour générer la population initiale de l'algorithme génétique. La population initiale doit contenir des chromosomes qui soient bien répartis dans l'espace des solutions pour fournir à l'algorithme génétique un matériel génétique varié. La façon la plus simple est de générer aléatoirement les chromosomes.

b) L'opérateur de sélection

La sélection tend à augmenter l'importance des bonnes solutions par rapport aux mauvaises .C'est une heuristique utilisée par l'algorithme génétique : les bonnes solutions sont supposées être les plus prometteuses pour la génération de descendants.

Il existe plusieurs méthodes de sélection. Les plus connues sont la sélection proportionnelle à la fonction fitness, la sélection sur le rang et la sélection en tournoi.

L'indice de qualité (fitness), aussi appelé indice de performance, est une mesure abstraite permettant de classer les chromosomes.

c) L'opérateur de croisement (crossover)

L'opérateur de croisement combine le matériel de un ou plusieurs parents pour obtenir un ou plusieurs enfants. Il existe différents types de croisement, nous allons brièvement présenter les trois principaux : Le croisement un point détermine aléatoirement un point de coupure et

échange la deuxième partie des deux parents. Le croisement deux points (qui peut être étendu à points) possède 2 points (ou) de coupures qui sont déterminés aléatoirement. Enfin le crossover uniforme échange chaque bit avec une probabilité fixé à ½.

d) L'opérateur de mutation

Le rôle de la mutation consiste à faire apparaître de nouveaux gènes. Cet opérateur introduit une diversité nécessaire à l'exploration de l'espace de recherche en permettant de générer des points dans des régions a priori sans intérêt .La mutation la plus simple sur un chromosome change un bit de façon aléatoire. Un chromosome a une probabilité de mutation d'un taux.

e) Remplacement

Cette dernière étape du processus itératif consiste en l'incorporation des nouvelles solutions dans la population courante. Les nouvelles solutions sont ajoutées à la population courante en remplacement (total ou partiel) des anciennes solutions. Généralement, les meilleures solutions remplacent les plus mauvaises ; il en résulte une amélioration de la population. Lorsque la nouvelle population n'est constituée que de nouvelles solutions, on parle d'algorithme génétique générationnel.

f) Avantages et inconvénients

D'abord, les algorithmes génétiques sont coûteux en temps de calcul, puisqu'ils manipulent plusieurs solutions simultanément. C'est le calcul de la fonction de performance qui est le plus pénalisant, et on optimise généralement l'algorithme de façon à éviter d'évaluer trop souvent cette fonction.

Ensuite, l'ajustement d'un algorithme génétique est délicat. L'un des problèmes les plus caractéristiques est celui de la dérive génétique, qui fait qu'un bon individu se met, en l'espace de quelques générations, à envahir toute la population. On parle dans ce cas de convergence prématurée, qui revient à lancer à une recherche locale autour d'un minimum… qui n'est pas forcément l'optimum attendu. Les méthodes de sélection proportionnelle peuvent en particulier favoriser ce genre de dérive. Un autre problème surgit lorsque les différents individus se mettent à avoir des performances similaires : les bons éléments ne sont alors plus sélectionnés, et l'algorithme ne progresse plus.

Le choix d'une représentation « intelligente » pour permettre un remplacement générationnel efficace est un autre aspect de la question, et l'efficacité d'un algorithme génétique dépend beaucoup de la façon dont on opère le croisement des individus. Ainsi, dans le cas du problème du voyageur de commerce, on peut envisager d'apparier selon la méthode

des coupures, telles que nous l'avons vu plus haut. Considérons les deux tournées suivantes, et opérons une coupure entre la troisième et la quatrième ville, puis inversons les gènes :

Parent 1 : A – F – B – C – E – D

Parent 2 : C – B – A – F – D – E

On obtient alors les successeurs :

Enfant 1 : A – F – B – C – D – E

Enfant 2 : C – B – A – F – E – D

Ici, des trajets initiaux, on n'a gardé que les trois premières villes (A – F – B), et on a placé dans l'enfant 1 les villes qui ne sont pas présentes dans la première partie (C – D – E), dans l'ordre où elles apparaissent dans le parent 2. On a fait le contraire pour l'enfant 2.

Mais on pourrait également recomposer en fonction du nombre d'adjacences de villes communes entre les deux parents. Si deux tournées possèdent des villes adjacentes en commun, il peut être intéressant que les trajets « enfant » les contiennent aussi. Des méthodes spécifiques de remplacement générationnel ont ainsi été élaborées, comme le « edge-3 » de Mathias et Whitley, qui ont montré leur grande efficacité. Mais il faut du temps pour se consacrer à l'étude et à la mise en place de bons opérateurs génétiques, face à un problème donné, et un algorithme génétique est sans doute plus délicat à faire fonctionner du premier coup qu'un algorithme du recuit simulé.

Le grand avantage des algorithmes génétiques est qu'ils parviennent à trouver de bonnes solutions sur des problèmes très complexes, et trop éloignés des problèmes combinatoires classiques pour qu'on puisse tirer profit de certaines propriétés connues. Ils doivent simplement déterminer entre deux solutions quelle est la meilleure, afin d'opérer leurs sélections. On les emplois dans les domaines où un grand nombre de paramètres entrent en jeu, et où l'on a besoin d'obtenir de bonnes solutions en quelques itérations seulement – dans les systèmes de régulation de transport en temps réel par exemple.

Par ailleurs, les algorithmes génétiques se prêtent bien, du fait de leur traitement simultané de solutions, à la recherche d'**optimum multiples** : en créant une fonction de coût partagée, dont la valeur dépend partiellement de la distance entre les individus, on voit se former graduellement des sous-populations d'individus, qui se stabilisent autour des différents pics de la fonction objectif. C'est la technique du inchangé par la méthode du partage[9].

2.3. Conclusion

Dans ce chapitre ,nous avons présenté les algorithmes bio-inspirés, d'une part ,les algorithmes essaim de particules, d'autre part, les algorithmes évolutionnaire.

Nous avons détaillé dans certains algorithmes, et nous sommes basés sur l'algorithme génétique, et leur operateur avec les avantages et l'inconvénient.

Notre intérêt à portée sur l'étude de l'application des algorithmes génétiques dans le domaine de bio-inspiration. Vu leur grande efficacité face aux problèmes afin de proposer un algorithme génétique pour le problème d'emploi du temps.

Chapitre 3

Résultat et discussion

3.1. Introduction

Le problème de l'emploi du temps est un problème représentatif d'une famille de problèmes combinatoires discrets. Il renferme un ensemble d'objectifs conflictuels, un ensemble de contraintes non linéaires et un nombre de combinaisons potentielles très élevé.

Dans ce chapitre, le problème de l'emploi du temps sera prouvé que lui aussi est un problème d'optimisation. Jusqu'à maintenant ce sont principalement les modèles mono-objectifs qui sont utilisés lors de la résolution de ce type de problèmes. Ainsi, nous allons concentrer notre travail dans ce qui suit sur l'utilisation de l'algorithme génétique, Pour la résolution de ce problème.

3.2. Description du problème à résoudre

Dans un établissement éducatif, un ensemble d'étudiants groupés sous une structure hiérarchique (département, cours, salles, promotions, groupes,…) sont censés avoir un ensemble d'enseignements qui se répètent périodiquement, chacun de ces enseignements s'étend sur une durée de temps, dont l'unité élémentaire est la période.

Résoudre le problème de l'emploi du temps revient à affecter à chacun de ces enseignements un nombre de périodes consécutives égal à la durée qu'il exige, un local dont le type et la capacité sont convenables, et un enseignant apte à assurer le module concerné par l'enseignement de façon à prévenir les conflits sur les enseignants, sur les étudiants et sur les locaux..

Le problème de l'emploi du temps se manifeste en plusieurs différentes formes dont chacune des formes est spécifique à l'environnement ou à l'institution qui en a besoin. Dans notre cas, le problème de l'emploi du temps étudié est celui d'université (département d'informatique) où les responsables pédagogiques ont besoin chaque année d'établir une nouvelle planification des différentes promotions en essayant au mieux de satisfaire les contraintes « humaines » des enseignants et des étudiants, les contraintes pédagogiques imposées par la progression des enseignements et en tenant compte des contraintes « physiques » liées aux ressources matérielles (les locaux, les équipements etc…).

Le département d'informatique regroupe différentes formations qui ont une durée qui varie entre trois ans (LMD) et cinq ans (MASTER2).

Le programme pédagogique de chaque formation est connu à priori. Ce programme précise les modules à suivre, leurs volumes horaires et quelques informations pédagogiques (répartition en cours, travaux dirigés, travaux pratiques etc…). Selon les besoins pédagogiques et les conditions physiques des ressources, chaque formation est structurée en promotions, en sections, et en groupes, le nombre d'étudiants par groupe en travaux dirigés(TD) est limité à 30 pour préserver un meilleur suivi des étudiants.

Les enseignants interviennent selon leur discipline et leur domaine de compétence. Administrativement, les enseignants doivent assurer un nombre minimal d'heures qui est défini dans leur statut. Lorsqu'un un enseignant est chargé d'un enseignement donné, il est tenu d'en respecter le volume horaire prévu par le responsable pédagogique. En cas d'absence, l'enseignant doit prévoir des séances de rattrapage. Il doit donc connaître précisément la

disponibilité des ressources de sa séance. Cette organisation garantit que tous les étudiants qui suivent une même formation auront eu le même volume horaire d'enseignement.

En résumé, les données du problème à résoudre sont constituées par :

1) un ensemble de créneaux horaires étalés sur une semaine de cinq jours, du dimanche au jeudi avec un nombre six périodes (du dimanche au mercredi) et un nombre de trois périodes le jeudi. La durée d'une période est d'une heure trente minutes.
2) Un ensemble de promotions ou groupes d'étudiants.
3) Un ensemble de cours, TD ou TP à programmer dans la semaine.
4) Un ensemble de locaux (salles, amphis et labos).

L'affectation des modules, enseignants, locaux à des périodes est soumise à des contraintes qui diffèrent selon leurs priorités (l'intérêt que recouvre la satisfaction d'une contrainte ou sa violation).

Une contrainte ne revêt pas nécessairement un aspect absolu (soit elle est vérifiée ou violée) mais peut être formulée sous forme d'un objectif qui doit être approché autant que possible, selon ce critère, les contraintes peuvent être réparties en deux grandes classes : les contraintes dures (absolues) et les contraintes de préférences.

3.2.1. Les contraintes dures

Ce type de contraintes doit être obligatoirement satisfait dans toutes les situations car la violation de l'une de ces contraintes rend l'emploi du temps inefficace dans la réalité. On distingue dans notre cas six contraintes dures :

1) Un enseignant ne peut pas être affecté à deux séances différentes à la même période.
2) Une salle ne peut pas accueillir deux séances différentes à la même période.
3) Chaque enseignant doit enseigner un module qui entre dans ses compétences.
4) Un module doit respecter le nombre de séances hebdomadaires de ce dernier c'est à dire si un module est enseigné trois fois par semaine, alors il doit apparaître 3 fois dans le chromosome de la promotion.
5) Un emploi du temps doit comporter tous les modules d'une promotion.
6) La charge journalière d'un enseignant ne doit pas être dépassée.

3.2.2. Les contraintes de préférence

Contrairement au type de contraintes précédent, les contraintes de préférences n'exigent pas la vérification stricte, mais d'approcher au maximum de l'objectif voulu. Dans notre cas, on distingue :

1) Essayer d'éviter aux (enseignant ou étudiants) des pertes de temps par de trop longs espacements entre deux séances d'une même journée (pas de trous).
2) Eviter que certains jours se trouvent surcharger alors que d'autres le sont moins.
3) Eviter d'affecter une période jugée non convenable à un enseignant, sauf si ce la est inévitable
4) Les modules de coefficient minimal ne doivent pas occuper les séances de la matinée d'une journée donnée, au détriment des modules de coefficients élevés.
5) Minimiser les déplacements des étudiants dans l'établissement.
6) Libérer quelques après-midi pour les enseignants

Afin de modéliser l'ensemble des contraintes régissant notre problème, on doit d'abord définir les structures de données nécessaires.

3.3. Structures de données

Dans notre cas, nous proposons les structures de données suivantes :

- Un tableau d'activités pédagogiques ACT comportant les différentes activités pédagogiques à programmer pour une promotion donnée. Le contenu de chaque élément est un enregistrement représentant l'activité pédagogique numéro i et rassemblant l'enseignant i1 et l'évènement i2 (un événement est composé d'un module en terme de cours, TD ou TP et le groupe (ou section) d'étudiants concerné par cet enseignement).

Ens1 even1	Ens1 Even1	Ens1 Even2	Ens2 Even2		Ens i Even j	…….	

- Un tableau événement EVEN comportant les différents enseignements à programmer pour une promotion donnée. Le contenu de chaque élément est un enregistrement représentant l'évènement i et rassemblant l'enseignement et les différents groupes d'étudiants concerné par cet enseignement.

M1g1	M1g1	M2g1	M2g3	……	migj	……

- Un vecteur binaire ETATE associé à chaque enseignant indiquant l'état de l'enseignant à la période i, chaque élément du vecteur est une valeur binaire :0 signifie que l'enseignant est libre, et 1 signifie que l'enseignant est occupé.

0	1	1	0	……	1	0	

- Un vecteur binaire ETATS associé à chaque salle indiquant l'état du salle à la période i, chaque élément du vecteur est une valeur binaire : 0 signifie que le salle est libre est libre, et 1 signifie que le salle est occupé.

1	0	0	1	……			0

- Une matrice emploi du temps EPDT représentant toutes les séances de la semaine (en colonnes), qui sont au nombre de 27 périodes allant du dimanche au jeudi et en ligne les différents locaux. EPDT(i, j)=k si l'activité pédagogique k est programmée à la période j dans la salle i.

Chaque élément de l'EPDT est un enregistrement comportant, l'enseignant, la salle et l'évènement impliqué dans la période correspondante. Ainsi pour concevoir l'emploi du temps, il faut remplir tous ses enregistrements tout en respectant l'ensemble des contraintes.

3.4. Pseudo code

```
private final ArrayList<ClassPRJ> classes;
private boolean isFitnessChanged = true;
private double fitness = -1;
private int classNumb = 0;
private int numbOfConflicts = 0;
private final Data data;

public Schedule(Data data) {
    classes = new ArrayList<>(data.getNumberofClasses());
    this.data = data;
}

public Schedule initialize() {
    new ArrayList<>(data.getDepartments()).forEach(dept -> {
        dept.getCourses().forEach(course -> {
            ClassPRJ newClass = new ClassPRJ(classNumb++, dept, course);
            newClass.setMeetingTime(data.getMeetingTimes().get((int) (data.getMeetingTimes().size() * Ma
            newClass.setRoom(data.getRooms().get((int) (data.getRooms().size() * Math.random())));
            newClass.setInstructor(data.getInstructors().get((int) (course.getInstructors().size() * Mat
            classes.add(newClass);
        });
```

3.5. Etude de cas

L'étude de cas est une méthode utilisée dans les études qualitatives en sciences humaines et sociales mais peut-être utiliser dans les études pour se pencher sur un cas en particulier. Elle vise l'étude approfondie d'un cas demandé, soit-il une personne ,un groupe ou un sujet spécifique.

3.5.1. Université Mohamed Boudiaf

L'établissement a débuté en 1985[28] par l'ouverture d'un institut d'enseignement supérieur en mécanique, suivi en 1989par ceux de génie civil et de gestion des techniques urbaines. En 1992, il est devenu centre universitaire et, en 2001, il a accédé au statut d'université[15] avec quatre facultés et 23 départements[18].Selon les statistiques de l'université pour l'année 2013, le nombre de facultés et d'instituts, a augmenté à 7 et 2respectivement. Cependant, le nombre de laboratoires de recherche au sein de l'établissement et qui sont agréés par le ministère de l'Enseignement supérieur et de Recherche scientifique a augmenté de7 à 23 laboratoires.

L'université de M'Sila peut accueillir 36 217 étudiants[26] et dispose de 10 000lits en résidence universitaire.

L'établissement porte le nom de Mohamed Boudiaf.

3.5.2. Département de l'informatique

L'institut d'informatique a été créé en septembre 1997.il a procédé ses activités par la formation de la 1éré promotion D.E.U.A en informatique de gestion et système.

- En 1999 inscription de la première promotion des ingénieurs, option système d'information.
- En 2003 inscription de la première promotion de formation en post graduation, option informatique industrielle ingénieurs.
- En 2007 le département informatique a adopté pour la première fois le nouveau système pédagogique L.M.D par l'inscription de 69 étudiants en 2éme année licence, spécialité informatique académique.
- Les perspectives du département informatique sont illimitées notamment :
- Ouverture d'autre formations L.M.D
- Ouverture du Master, organisation des séminaires, journées, colloques scientifiques, dans le but promouvoir les compétences de l'ensemble des enseignants [33].

3.5.3. Résultat de l'étude de cas

a) NetBeans

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL(Common Development and Distribution License) et GPLv2. En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML,le Groovy, le PHP et le HTML, ou d'autres (dontPython et Ruby) par l'ajout de greffons. Il offre toutes les facilités d'un IDE moderne (éditeur en couleurs, projets multi-langage, refactoring ,éditeur graphique d'interfaces et de pagesWeb) [34].

b) Définitions des classes

- Gestion des départements : cette classe est pour ajouter, modifier ou supprimer un nom de département.

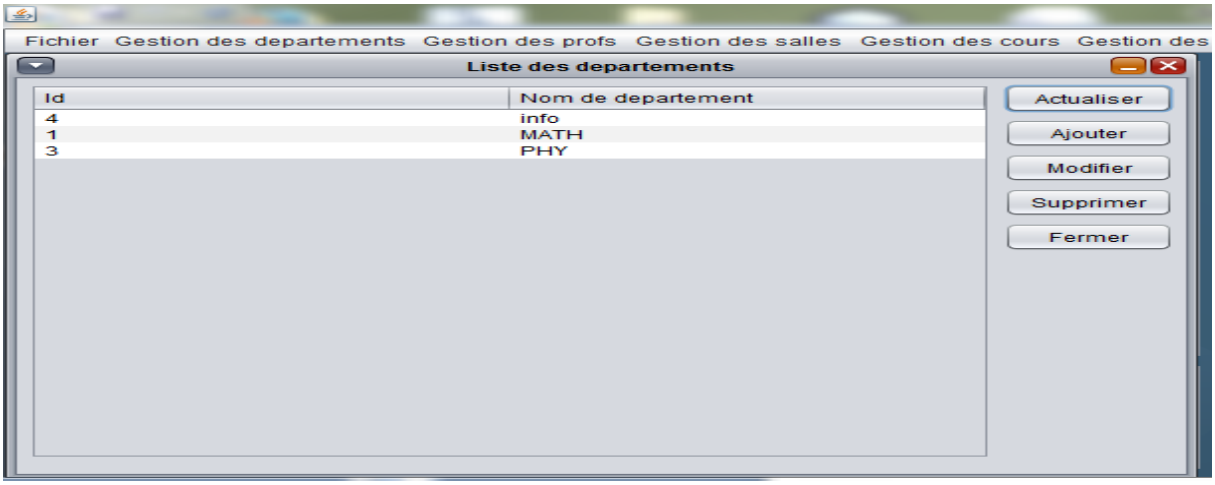


Figure 3.1 : gestion de département

- Gestion des professeurs :cette classe pour insérer les noms des professeurs, les modifier ou les supprimer.

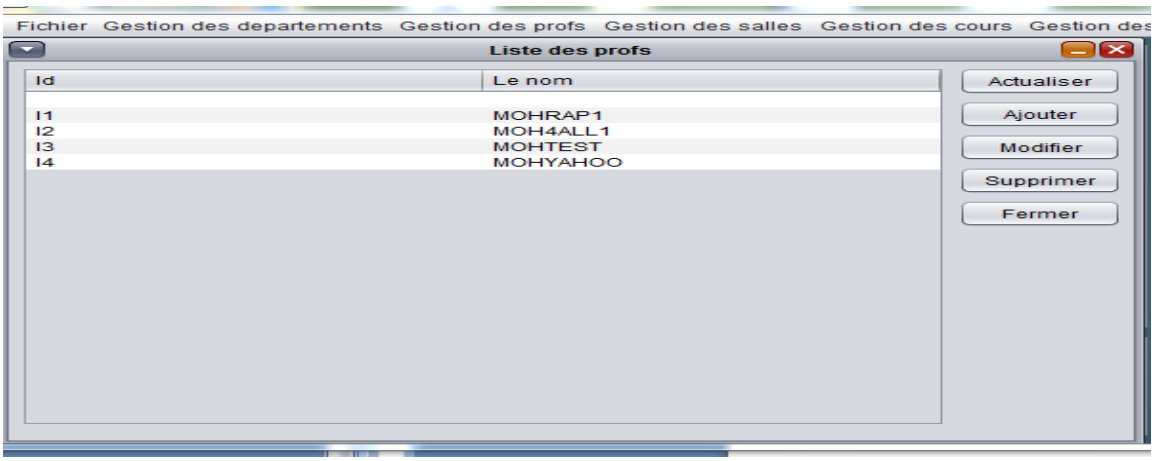


Figure 3.2 :gestion des professeurs

- Gestion des salles : cette classe pour insérer les capacités et les numéros des salles et leurs types, et on peut modifier ou supprimer.

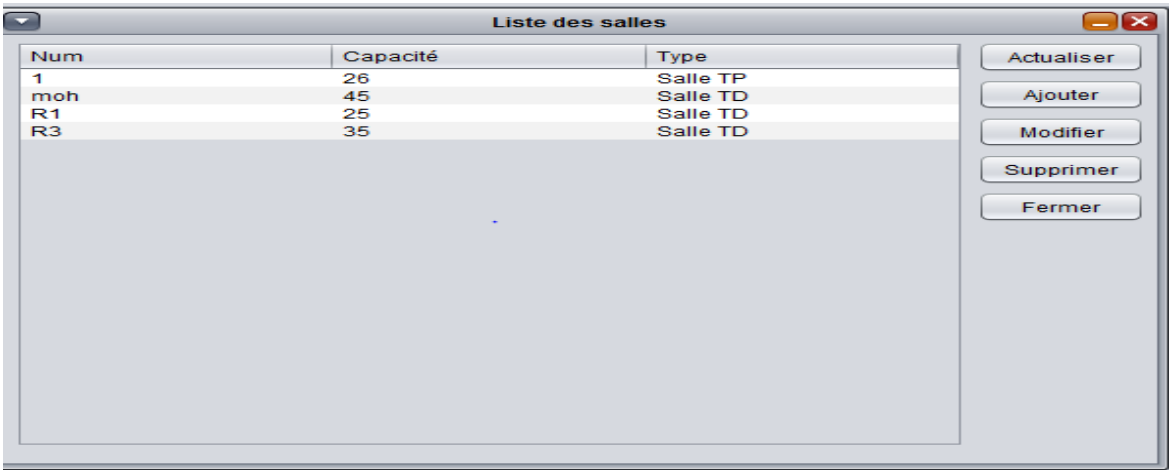


Figure 3.3 :gestion des salles

- Gestion des cours : C'est une classe qui sert à insérer les noms des cours et le nombre maximum des étudiants, avec la possibilité de modifier ou supprimer.

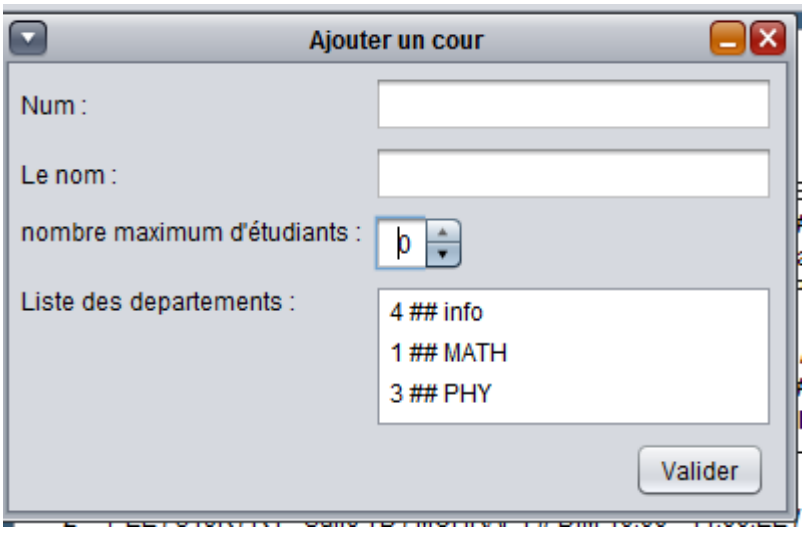


Figure 3. 4 :gestion des cours

- Gestion de temps : cette classe pour insérer le temps, avec la possibilité de modifier ou supprimer.

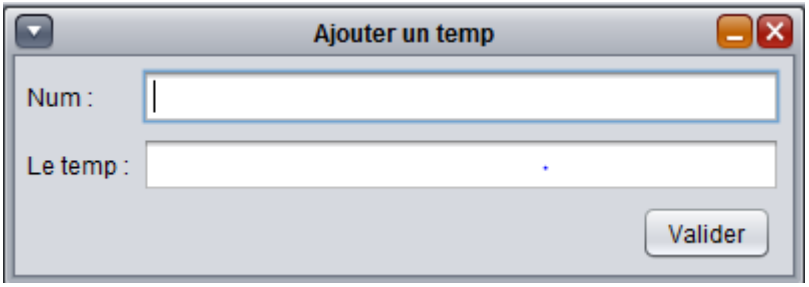


Figure 3.5 :gestion de temps

Après notre étude appliquée à un cas générale, les résultats obtenus sont comme le suivant :

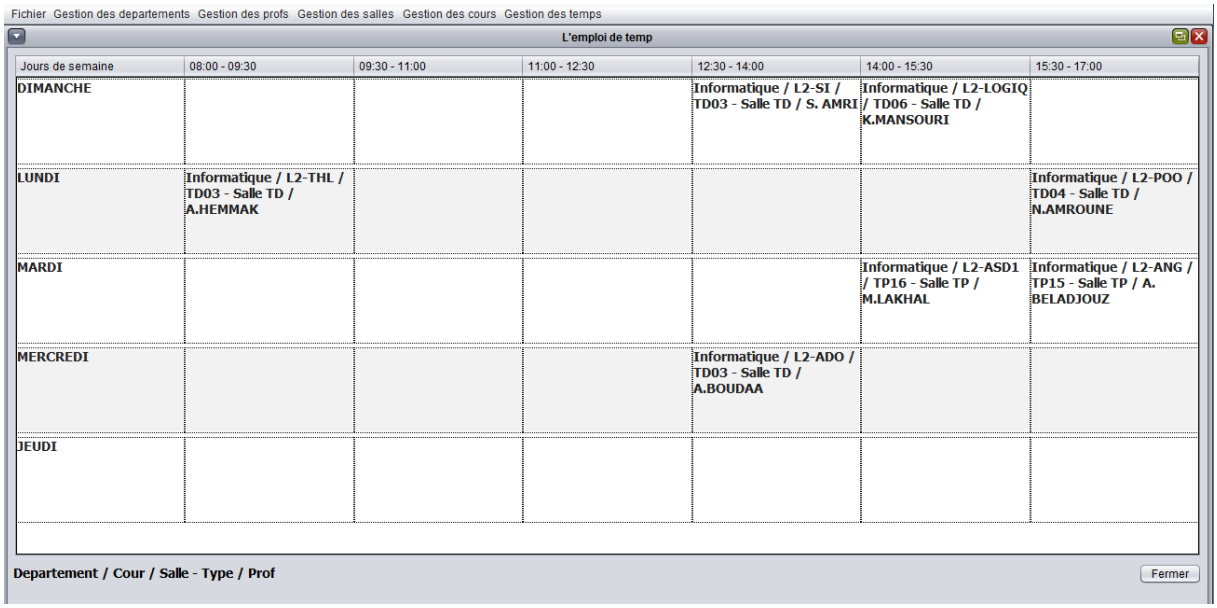


Figure 3.6 : emploi de temps

3.6. Conclusion

Dans ce chapitre, nous avons utilisés l'algorithme génétique pour résoudre le problème d'emploi du temps, seuls les tests pratiques peuvent prouver l'efficacité d'une approche donnée. Cependant, par contrainte de temps insuffisant, l'étude d'un cas particulier n'était pas possible.

Conclusion générale et Perspectives

L'objectif de ce travail a été l'étude des approches de résolution des problèmes d'optimisation combinatoire et plus particulièrement le problème de l'emploi du temps afin de proposer une approche qui peut contribuer à la résolution de ce problème.

L'accent a particulièrement été mise sur l'utilisation des algorithmes bio-inspiré. Un choix justifié par la nature même du problème qui en réalité formulé sous forme d'un ensemble d'objectifs qui peuvent éventuellement être contradictoires.

L'objectif de cet algorithme bio-inspiré est de produire une solution efficace, l'algorithme proposé stocke la meilleure solution trouvée durant la recherche.

Dans les algorithmes génétiques ceci est réalisé en maintenant une population supplémentaire.

Un volume important de travail reste à faire. Premièrement ,l'implémentation de ce travail, ce qui permettra d'analyser les performances de la méthode. Les algorithmes génétiques sont souvent critiqués pour leur lenteur.

Les algorithmes génétiques sont aussi connus pour leur sensibilité quant au choix de la population initiale.

Références Bibliographiques

[1] Babes M., Ounas H., Elaboration d'un algorithme génétique pour résoudre le problème d'emploi du temps d'université, Université Badji Mokhtar Annaba, Algerie.Grenoble.

[2] Bergh V. d., Engelbrecht F. A. P., A Cooperative Approach to Particle Swarm Optimization, Transactions on Evolutionary Computation, vol. 8, issue 3, pp. 225– 239, 2004.

[3] Burke E., kingston j., Jackson k., Weare R., Automated university Timetabling: the state of the art, the Computer Journal 40 (9) 565-571, 1997

[4] Calas G., Optimisation par Essaim Particulaire, Technical report, Ecole pour l'Informatique et les Technique Avancées (EPITA), Paris, France, 2009

[5] Chan Y. C. P., La planification du personnel : acteurs, actions et termes multiples pour une planification opérationnelle des personnes, Thèse de doctorat, InstitutIMAG, Université Joseph Fourier-Grenoble, 1 octobre 2002.

[6] Chuang L. Y., Tsai J. H., Yang C. H., Binary Particle Swarm Optimization for Operon Prediction, Nucleic Acids Research, vol. 38, issue 12, pp. e128, 2010..

[7] El Dor A., Perfectionnement des Algorithmes d'Optimisation par Essaim Particulaire : Applications en Segmentation d'Images et en Electronique, thèse, Université Paris-Est, 2012.

[8] Esmin A. A., Coelho R. A., Matwin S., A Review on Particle Swarm Optimization Algorithm and its Variants to Clustering High-dimensional Data, Artificial Intelligence Review, pp. 1–23, 2013.

[9] Holland, Adaptation in Natural and Artificial Systems. University of Michigan Press Ann Arbor, 1975.

[10] Georges W., Heus K., Patrice François, « Gymnaste : Aide à l'élaboration des roulements infirmiers. Du traitement des absences au management participatif », Laboratoire TIMC, SILM , CHU de Grenoble , Université Joseph Fourier- Grenoble,1994.

[11] Guangyou Y., A Modified Particle Swarm Optimizer Algorithm, 8th International Conference on Electronic Measurement and Instruments (ICEMI), pp. 318– 321, 2007.

[12] Khanesar M. A., Teshnehlab M., Shoorehdeli M. A., A Novel Binary Particle Swarm Optimization, IEEE Mediterranean Conference on Control & Automation (MED), pp. 1–6, 2007.

[13] Kennedy J., Eberhart R., A Discrete Binary Version of the Particle Swarm Algorithm, in IEEE International Conference on Systems, Man, and Cybernetics, vol. 5, pp. 4104–4108, 1997.

[14] Kennedy J., Eberhart R. C., Particle swarm optimization , proceeding of the IEEE conference on neural networks, IV, Piscataway, NJ, pp. 1942-1948, 1995.

[15] Merzouk S. E., Problème de dimensionnement de lots et de livraisons : application au cas d'une chaîne logistique, Thèse de doctorat de l'Université de Technologie de Belfort- Montbéliard et de l'Université de Franche-Comté, Novembre 2007.

[16] Nouaouria N., Boukadoum M., Proulx R., Particle Swarm Classification: A Survey and Positioning, Pattern Recognition, vol. 46, issue 7, pp. 2028–2044, 2013.

[17] Prabhneet k., Taranjot k., A Comparative Study of Various Metaheuristic Algorithms, Guru Teg Bahadur Institute of Technology, GGSIPU, New Delhi

[18] « Présentation de l'université » , sur http://www.univ-msila.dz (consulté le 14 /04/ 2018).

[19] Remy-R, Alexander-J, «Systèmes interactifs d'aide à l'élaboration de plannings de travail de personnel», Thèse de doctorat ,Laboratoire TIMC ,Instutit IMAG , Université Joseph Fourier-Grenoble,07 novembre 2003 .

[20] Romana C. H., Adis A., Milan TUBA,: 978-960-474-330-8.

[21] Saharan L. K., Mittal M. L., Comparison of Various PSO Variants with Different Mathematical Functions, International Journal on Theoretical and Applied Research in Mechanical Engineering, vol. 2, issue 3, pp. 135–142, 2013.

[22] Sandhu k., Automating class schedule generation in the context of university timetabling information system , School of Management, Nathan Campus, Griffith University, 21 September 2001.

[23] Saoucha A., Ghanem N, K., Benmammar B., On applying firefly algorithm for cognitive radio networks. Communications and Mirjana VUKOVIC, Milenko PIKULA, "Firefly Algorithm for Constrained Optimization Problems". ISBN Vehicular Technology in the Benelux (SCVT), 2014 IEEE 21st Symposium on. IEEE, 2014.

[24] Saoucha N. A., Paramétrage des algorithmes génétiques pour l'optimisation de la QoS dans les réseaux radios cognitifs, thèse de magistère, Université de M'sila, mars 2013.

[25] Schaerf A., Schaerf M., Local search techniques for large high school timetabling, in proceeding of the 1st international conference on the practice and theory of automated timetabling, pp. 313-323, 1995.

[26] « Statistiques 2013-2014 » , sur http://www.univ-msila.dz (consulté le 16 décembre 2013)

[27] Troudi-F , « Résolution de problème de l'emploi du temps :propose un algorithme multi objectif »Thèse de magister en informatique ,Université Mentouri , Constantine 2006

[28] « University of M'sila (Mohamed Boudiaf) » , sur http://uniandi.com (consulté le 16 /04/2018)

[29] Van den Bergh ,F .,Engelbrecht ,A.P.:Cooperative Approach to Particle Swarm optimization, Transaction on Evolutionary Computation, vol.8, issue 3 ,pp.225-239,2004.

[30] Wang L., An Improved Cooperative Particle Swarm Optimizer, Telecommunication System, vol. 53, issue 1, pp. 147–154, 2013.

[31] Werra D., An Introduction to Timetabling, European Journal of Operational research 19, 151-162.1985

[32] wikipedia, www.wikipedia.bio-inspiration.com, consulté le 16 /05/2018

[33] wikipedia, www.univ m'sila.com, consulté le 16 /03/2018

[34] wikipedia, www.wikipedia.netbeans.com, consulté le 14 /04/2018

[35] Yang X. S.,Firefly algorithm for multimodal optimization in proceedings of the stochastic Algorithms", (SAGA 109) vol.5792, Oct.2009.

[36] Yang X. S., Nature-Inspired Metaheuristic Algorithms. Luniver Press, UK. 2008.

[37] Yang X.S., Nature-Inspired Metaheuristic Algorithms, 2nd Edition Copyright © 2010 Luniver Press.

[38] Yang X. S., Firefly algorithm, levy flights and global optimization, in Research and Development in intelligent systems XXVI. Springer, 2010,pp.209-218.

[39] Yang X. S., Firefly algorithm, stochastic test functions and design optimization,International Journal of Bio-Inspired computation, vol. 2, no. 2, pp. 78-84, 2010.

[40] Zhan, Z. H., Zhang J. Li. Y., Chung H. H., Adaptive Particle Swarm Optimization, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 39, issue 6, pp. 1362–1381, 2009.

Résumé

Cette dissertation offre les résultats d'une enquête, et l'optimisation du système de planification à l'Université de Msila (en tant que étude de cas) en utilisant l'algorithme génétique (GA). Les Techniques GA sont utiles pour résoudre le problème d'ordonnancement du monde réel tel que le calendrier qui est un travail complexe et généralement fait manuellement.

Ce travail se concentre sur l'horaire des cours avec attribution des événements (temps, sujet et conférencier) dans une manière appropriée en utilisant la ressource disponible et aide à éviter les conflits. Les algorithmes explorent des différents opérateurs de GA comme le croisement, la mutation et le mécanisme de sélection qui est appliqué à l'ensemble des chromosomes. Les tests ont été produits en utilisant les différents paramètres de l'emploi de temps, croisement, et la probabilité de mutation. Croisements à deux points mis en œuvre au calendrier pour obtenir la solution optimale en utilisant diverses probabilités de croisement. Ce travail recommande d'améliorer la fonction de remise en forme et utiliser un mécanisme de sélection différent de l'algorithme.

Mots clés : Algorithme génétique, algorithme bio-inspiré, emploi de temps

Abstract

This dissertation offers the results of an investigation, and optimization of scheduling system in M'sila University (as a case study) using the genetic algorithm (GA). GA techniques are useful for solving real-world scheduling problem such as timetable which is a complex work and usually done manually.

This work focuses on scheduling courses timetable to allocate events (time, subject, and lecturer) in an appropriate way by using the available resource and assists to avoid conflicts. The algorithms explored different operator of GA such as crossover, mutation, and selection mechanism that's applied to set of chromosomes. The testing has been produced using different parameters of timetable, crossover, and mutation probability. Two point crossovers implemented to the timetable to obtain the optimal solution using various probabilities of crossover. This work recommended to enhance the fitness function and used different selection mechanism to the algorithm.

Keywords:genetic algorithm, bio-inspired, schedule problem.

ملخص

تقدم هذه الرسالة نتائج التحقيق ، تحسين نظام الجدولة في جامعة المسيلة (كدراسة الحالة) باستخدام الخوارزمية الجينية (AG). تقنيات AGمفيدة لحل مشكلة الجدولة في العالم الحقيقي مثلالجدول الزمني الذي هو عمل معقد وعادة مايتم ذلكيدويا.

يركز هذا العمل على جدولة مواعيد الدوراتلتخصيص الأحداث (الوقت والموضوع والمحاضر) فيالطريقة المناسبة باستخدام الموارد المتاحة وتساعد علىتجنب الصراعات. الخوارزميات استكشاف مختلف المشغل AG مثل التبادل، والتحول، وآلية الاختياريتم تطبيقه على مجموعة من الكروموسومات. كان الاختبار المنتج باستخدام معايير مختلفة للجدول الزمني، واحتمالية الطفرة. اثنين من عمليات الانتقالتنفذ للجدول الزمني للحصول على الحل الأمثلباستخدام احتمالات مختلفة من التبادل. هذه العمل أوصت لتعزيزوظيفة اللياقة واستخدام آلية اختيار مختلفة للخوارزمية

كلمات البحث :.خوارزمية الجينُية، الخوارزميات المستوحاة من الطبيعة، الجدول الزمني.