

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche
scientifique

UNIVERSITE SAIDA- Dr. MOULAY Tahar
FACULTE :TECHNOLOGIE
DEPARTEMENT :INFORMATIQUE



MEMOIRE DE MASTER

OPTION : MICR

THEME

***Approche bio inspirée pour la résolution du
problème de covoiturage régulier***

Présenté par :

Dahak Souhila
Ameur Moussa Lakhdar

Encadré par :

Dr. Bendaoud Zakaria

Promotion : 2019/2020

REMERCIEMENTS

Au terme de ce travail , je tiens à remercier toutes les personnes ayant contribué à la réalisation de ce mémoire.

Un remerciement particulier à est adressé à Monsieur **Z. Bendaoud** pour avoir proposé, suivi et dirigé le travail. je tiens à lui exprimer ma sincère gratitude pour ses conseils pertinents, pour ses orientations constructives et pour le temps si précieux qu'il m'a consacré.

Mes vifs remerciements s'adressent aussi aux membres du jury Monsieur

pour leur disponibilité et acceptation d'examiner et de rapporter mon travail.

Dédicace

A mon père et ma mère

Pour l'amour qu'ils ont su me prodiguer

A mes frères et mes grands-parents

Pour leur soutien et leur patience

A mes nièces

Pour le bonheur qu'elles apportent à ma vie

A moi

Preuve de résistance et de patience .

Dahak souhila

Dédicace

A mon père et ma mère

Pour l'amour qu'ils ont su me prodiguer

A mon frère et ma sœur

Pour leur soutien et leur patience

A mes nièces

Pour le bonheur qu'elles apportent à ma vie

A moi

Preuve de résistance et de patience .

Ameur Moussa Lakhdar

ملخص

أصبح استخدام مشترك للسيارات على مدار السنوات العشر الماضية أسلوباً جديداً للتنقل للملايين الأشخاص. وقد أصبح وسيلة شائعة جداً للسفر. و تتمثل في الاستخدام المشترك للسيارة من قبل سائق غير محترف وواحد (أو أكثر) لراكب (ركاب) لغرض القيام برحلة مشتركة كلها أو جزء منها ، يقلل هذا الأسلوب من تكاليف النقل من خلال قسمة تكلفة الرحلة على عدد الركاب ويقلل من حركة المرور و التلوث. كما أنه يساعد في استعادة بعض الاتصالات التي اختفت في وسائل النقل العام. نحن مهتمون بإنشاء نظام مُحسَّن لاستخدام السيارات المشتركة ، فهذه المشكلة لها تعقيد ليس أقلها ، وكل جهودنا موجهة لإحياء هذه العقبة ولتنفيذ تطبيق برمجي تنافسي على نطاق واسع تقديم الرضا وجودة الخدمة. للقيام بذلك ، قمنا بتكييف ميتاوريسيتية مستوحاة من سلوك اليراعات الطبيعية لحل هذه المشكلة. يتم إنشاء ذلك على شبكة جغرافية تمثل مناطق تموضع الأشخاص المسجلين في التطبيق ، حيث يكون الهدف هو العثور على أفضل فرد في كل مجموعة فرعية والتي ستكون نقطة التقاء الأعضاء الآخرين للمجموعة الفرعية. من أجل القيام برحلة مشتركة .

Résumé

Le covoiturage est devenu ces 10 dernières années un nouveau réflexe de déplacement pour des millions Person .C'est devenu une façon de voyager très populaire .il consiste en l'utilisation commune d'un véhicule par un conducteur non professionnel et un (ou plusieurs) passager(s) dans le but d'effectuer tout ou une partie d'un trajet commun.Ce mode de déplacement permet de diminuer les frais de transport en divisant le coût du trajet par le nombre de passagers et de réduire le trafic et la pollution. Il permet aussi de restaurer une certaine communication qui a disparu dans les transports en commun. nous nous intéressons à l'établissement d'un système de covoiturage optimisé Ce problème ayant une complexité qui n'est pas des moindres, tous nos efforts sont dirigés dans le but de contrecarrer cet obstacle et mettre en œuvre une application logicielle compétitive à grande échelle offrant satisfaction et qualité de service. Pour ce faire, nous avons adapté une métaheuristique inspirée du comportement des lucioles naturelles pour résoudre ce problème. Celle-ci est établi sur un réseau géographique qui représente les zones de concentration des personnes inscrit dans l'application, où le but consiste à trouver le meilleur individu dans chaque sous-groupe qui va être le point de convergence des autres membres du sous-groupes afin d'effectuer un trajet en commun.

Abstract

Carpooling has become over the past 10 years a new commuting reflex for millions of people. It has become a very popular way of traveling. It consists of the joint use of a vehicle by a non-professional driver and one (or more)) passenger (s) for the purpose of making all or part of a common journey.This mode of travel reduces transport costs by dividing the cost of the trip by the number of passengers and reduces traffic and pollution. It also helps restore some communication that has disappeared in public transport. We are interested in establishing an optimized carpooling system This problem having a complexity which is not the least, all our efforts are directed in order to thwart this obstacle and to implement a competitive software application on a large scale offering satisfaction and quality of service. To do this, we have adapted a metaheuristic inspired by the behavior of natural fireflies to solve this problem. This is established on a geographic network that represents the areas of concentration of people registered in the application, where the goal is to find the best individual in each sub-group which will be the point of convergence of the other members of the sub-group. groups in order to make a joint trip

Table des matières

Table des matières

Table des figures

Liste des tableaux

Introduction générale	10
I. Etat de l'art sur le problème de covoiturage	12
I.1.Introduction	12
I.2.Historique	14
I.3. La voiture partagée	15
I.4. Autopartage	15
I.5. Covoiturage	16
I. 5.1 . Définitions.	16
I. 5.2.Classification du problème de covoiturage	17
I. 5.3 .TYPES DE DÉPLACEMENTS ET DE COVOITURAGE.....	19
I.6. La différence entre l'autopartage et le covoiturage	19
I.7.Heuristiques.....	20
I.8. Métaheuristiques	21
I.9. Conclusion.....	22
II. L'optimisation et les métaheuristiques	23
II.1. Introduction.....	23

II.2. Optimisation.....	24
II.2.1. Optimisation : Concept et définition.....	24
II.2.2. Délimitation de l'espace de recherche.....	24
II.2.3. Différents critères d'optimalité pour différents types d'optimum.....	24
II.2.4. Classification des problèmes d'optimisation.....	25
II.2.5. Optimisation mono-objectif.....	25
II.2.6. Optimisation multi-objectif.....	25
II.2.7. Les méthodes d'optimisation.....	26
<i>II.2.8. Les méthodes d'aide à la décision.....</i>	<i>29</i>
<i>II.3. Les Algorithmes Métaheuristiques Bio-Inspirés.....</i>	<i>29</i>
<i>II.3.1. Le bio-mimétisme.....</i>	<i>30</i>
<i>II.4. Optimisation dans les problèmes de covoiturage.....</i>	<i>31</i>
II.5.1 Complexité du problème de covoiturage dynamique optimisé.....	31
II.5.2. Avis et mesures.....	32
II.6. Les systèmes multi-agents au service de l'optimisation.....	32
II.6.1. SMA : le concept.....	32
II.6.2. Interaction et communication.....	35
II.6.3. Les techniques multi-agents intégrées au concept de covoiturage.....	36
II.7.1. Optimisation dans le covoiturage dynamique.....	36
II.8. Les algorithmes génétiques.....	36
II.9. Conclusion.....	41

Algorithme Firefly (luciole) pour la résolution du problème de covoiturage régulier **42**

III.1. Introduction.....	42
III.2. Algorithme des Lucioles (FireFly Algorithm).....	43
III.2.1. Inspiration.....	43

III.2.2. Principe de fonctionnement de l'algorithme des Lucioles.....	44
III.2.3. Génération de la population initiale.....	46
III.2.4. Fonction d'évaluation.....	47
a) Une fonction d'évaluation mono critère.....	47
b) Une fonction d'évaluation multicritère.....	48
III.2.5. Classement.....	49
III.2.6. Déplacement et mise à jour.....	49
III.2.7. Critère d'arrêt.....	50
III.3. Paramétrages des algorithmes Firefly.....	50
III.3.1. Nombre de luciole.....	50
III.3.2. Attraction de luciole.....	51
III.3.3. Attraction initiale.....	51
III.3.4. La distance.....	51
III.3.5. Coefficient d'absorption.....	52
III.3.6. Nombre de génération.....	52
III.4. Fonction objective.....	52
III.4.1. Avant déplacement.....	53
III.4.2. Après déplacement.....	53
III.5 .Vue artificiel VS Vue réelle.....	54
III.6 . Benchmarks.....	54
III.7 . Simulation et Résultat.....	55
III.8 .Conclusion.....	55
Présentation de l'application	56
conclusion générale	61
Bibliographie	62

Table des figures

I.1 La voiture partagée : Autopartage ou Covoiturage ?(1).....	15
I.2 Partage du véhicule.....	17
I.3 un exemple de DCCP (a 3).....	18
I.4 un exemple de LTCPP (a 3).....	18
I.5 principes de la métaheuristique basée sur la trajectoire (a 3).....	21
II.1 Classification des problèmes d'optimisation (19)	25
II.2 Optimisation multi-objectif et optimalité au sens de Pareto.....	26
II.3 Méthodes d'optimisation mon objectif (56)	26
II.4 Taxonomie des algorithmes méta-heuristiques bio-inspirés.....	30
II.5 Cycle de vie d'un agent (2).....	34
II.6 Les interactions sous leurs différentes formes.....	35
II.7 Sélection par la méthode de la roulette de casino (14).....	38
II.8 Les divers croisements binaire (40).....	39
II.9 Pseudo code de l'algorithme génétique (50).....	41
III.1 Les Lucioles.....	43
III.2 Organigramme de fonctionnement d'un algorithme FireFly (22).....	45
III.3 L'algorithme FireFly (19)	46
III.4 Fonctions mono critère et multi critère(21).....	48
III.5 Déplacement des lucioles dans une itération(21).....	49

Liste des tableaux

III.1 Vue réelle vs Vue artificielle	54
III.2 Résultats expérimentaux les individus.....	55

Introduction générale

Dans l'optique de répondre aux besoins d'innovation tout en demeurant dans un cadre respectueux des conditions environnementales fondamentales pour un climat sain, la voiture partagée se présente comme la solution clé pour une évolution sans conséquences néfastes en plus d'être d'ordre économique très avantageux.

Les outils mettant en place de tels services font ainsi l'objet d'intérêt de plusieurs industriels aussi bien que des chercheurs. Depuis plusieurs décennies déjà, le thème de l'innovation propre est placé en pleine ligne de mire par les gouvernements à l'échelle internationale vu la menace que traîne avec lui tout succès technologique atteint. En effet, sous l'ombre de l'avancement réalisé sur le plan du modernisme se cachent des méfaits et impacts péjoratifs majeurs touchant la qualité de la vie ainsi que l'équilibre biologique et environnementale nécessaire à la survie des êtres. Sous la lumière des sommets mondiaux nommés « sommets de la terre » dont les thématiques principales comprennent entre autre le développement durable comme point d'intérêt majeur, plusieurs études et travaux de recherche ont épuisé leurs efforts à faire émerger des énergies renouvelables et des technologies vertes. Dans cette lancée, des efforts ont été déployés pour atténuer l'impact négatif de la voiture personnelle : acteur pivot dans les taux d'émission de CO₂ et gaz à effet de serre. Les émissions de dioxyde de carbone sont d'une ampleur remarquable et vont crescendo avec l'évolution technologique; elles constituent de ce fait un facteur loin d'être négligeable dans la dégradation des conditions de vie.

Afin de soigner l'image environnementale du véhicule particulier, les systèmes d'Ecopartage sont nés. Dans ce contexte, le covoiturage en particulier a connu un succès notable grâce aux contributions qu'il apporte réduisant principalement le nombre de voitures en circulation. En effet, faisant de la voiture personnelle un mode de transport en commun, le covoiturage participe amplement à la réduction des taux d'émission de gaz nuisibles. Ses apports sont ainsi quantifiables en termes de CO₂ « non émis », conjointement aux multiples avantages qu'il offre aussi bien sur le plan individuel que collectif (e.g. réduction des budgets alloués au transport, flexibilité spatio-temporel, confort, équilibre social, etc.).

Plusieurs études ont ainsi été menées dans ce sens puisant dans les domaines de l'informatique, de l'intelligence artificielle, des GIS (Geographical Information Systems), de l'Internet ainsi que des domaines des télécommunications, etc.

Usant des technologies nouvelles, la mise en place d'un système plus ou moins évolué prime sur tout autre objectif au niveau des approches existantes. Celles-ci se sont en effet, pour la plupart, penchées sur l'adoption de la technologie (i.e. réseau bluetooth, intégration des PDAs, etc.) et délaissent les aspects d'efficacité des traitements (i.e. automatisation, optimisation, temps réel, aide au déplacement, etc.).

Ce travail consiste à utiliser une métaheuristique inspirée de la nature en particulier par l'intelligence en essaim des lucioles, (FireFly algorithm) pour résoudre le problème de covoiturage régulier.

Notre mémoire est organisé en trois chapitres plus une introduction et une conclusion générale.

Chapitre 1 : Dans le premier chapitre on présente une entrée sur la description du problème de covoiturage régulier.

Chapitre 2 : Dans le deuxième chapitre, on présente un état de l'art sur les méthodes d'optimisations, et une vue d'ensemble sur les algorithmes d'optimisation sont présentés.

Chapitre 3: Dans le troisième chapitre, nous exposons en détail les différentes étapes qui constituent la structure générale d'un algorithme Firefly.

Dans la deuxième partie, nous exposons la mise en œuvre de l'application qui a été faite, ainsi que les expérimentations et les résultats obtenus dans ce contexte.

Dans la deuxième partie, nous exposons en détail la mise en œuvre de l'application qui a été faite, ainsi que les expérimentations et les résultats obtenus dans ce contexte.

Enfin, une conclusion générale est présentée avec quelques perspectives proposée

I . CHAPITRE I

Etat de l'art sur le problème de covoiturage

I.1.Introduction

De nos jours, avec l'augmentation de la population et la dispersion de l'habitation, Le service de transport public est souvent incapable de servir efficacement les zones où des systèmes de transport rentables ne peuvent être mis en place. En conséquence, de plus en plus de gens utilisent les véhicules privées pour leur transport quotidien. Cependant, l'utilisation élevée de ces véhicules privées combinée à une mobilité humaine accrue augmentent la charge sur l'environnement et soulèvent des problèmes de transport tels que la congestion, le problème de stationnement et le temps de déplacement qui devient très conséquent.

Partage, convivialité, économies ,les arguments en faveur du covoiturage ne manquent pas. Grâce à Internet et au téléphone mobile, cette pratique, permettant de mettre en relation conducteurs et passagers, fonctionne et s'étend partout dans le monde, auprès de toutes les franges de la population. Le covoiturage, par opposition au fait de se déplacer seul dans sa voiture, offre plusieurs avantages. Ces avantages sont environnementaux, sociaux ou consistent simplement en la réduction du nombre De voitures sur les routes.

Les avantages financiers sont évidents, si deux personnes partagent une voiture pour un voyage, elles partagent également les coûts du carburant.

Si plus d'une personne monte dans une voiture, le nombre de voitures utilisées est par conséquent réduit, et par la même la occasion la menace pour l'environnement aussi.

L'étalement urbain est à l'origine de la croissance continue du trafic automobile qui induit une dégradation de la qualité de l'air, des nuisances sonores importantes, une trop grande utilisation de l'espace public, un gaspillage énergétique et une augmentation des émission de gaz à effet de serre . Dans le contexte de la transition énergétique, il convient de réduire ces impacts en agissant sur l'ensemble des déplacements et en particulier ceux contraints, tels les déplacements du quotidien domicile-travail, domicile-étude, etc. Dans de nombreux environnements, et en particulier dans les secteurs ruraux et périurbains où la densité est souvent trop faible pour mettre en place une offre de transports collectifs performante, le covoiturage apparaît comme une alternative efficace à l'auto-solisme et une importante source d'économie pour les ménages.

Les premiers services de covoiturage organisés ont été lancés dans les années 1980 par le milieu associatif. Depuis 2000, ces associations se sont pour la plupart professionnalisées, industrialisant leurs outils et adaptant leur offre pour proposer des solutions , faisant ainsi naître un métier nouveau : la prestation de service de covoiturage.

Après plusieurs années de développement rapide, le covoiturage a déjà été considéré comme un important service de transport alternatif à travers le monde. Afin de réduire le trafic et

d'encourager la mise en commun de la voiture, certains pays ont introduit une occupation élevée de véhicule (HOV) où seuls les véhicules avec deux passagers ou plus sont autorisés à conduire. Dans certains pays, il est également fréquent de trouver des places de stationnement spécialement réservées aux covitureuses. Beaucoup d'entreprises et d'autorités locales ont mis en place des systèmes de regroupement de voitures, souvent dans le cadre de programmes de transport plus larges.

I.2.Historique

Le covoiturage apparaît aujourd'hui comme étant une innovation sociale, une pratique nouvelle, développée par les nouvelles technologies. Il s'agit en réalité d'une pratique ancienne. Il

est possible de faire un lien entre le covoiturage et l'autostop, et faire remonter leur origine commune aux hobos (vagabonds saisonniers) américains qui circulaient gratuitement sur les trains de marchandise, puis dans les camions de marchandise. Le covoiturage a été popularisé pendant les conflits, notamment lors de la réquisition des taxis parisiens en 1914 (les fameux taxis de la Marne), ou grâce aux incitations anglaises¹⁵ et américaines¹⁶ entre 1939 et 1945. Dans le premier cas, il s'agit de déplacer rapidement un maximum de soldats dans une vision localisée. Dans le second l'objectif est d'économiser le carburant dans une vision globale à l'échelle d'un pays. Dans les années 1960-1970, les enfants du baby-boom pratiquent l'autostop à large échelle, pour des raisons pratiques, mais aussi culturelles et politiques (contre-culture, contestation de la société de consommation). En 1958, ALLOSTOP, la première association « d'autostop organisé » est créée pour mettre en relation, via des opérateurs téléphoniques, des conducteurs et des passagers. Cette institutionnalisation peut apparaître toutefois dans le contexte de l'époque comme étant une forme de déviance (Becker, 1985) vis-à-vis de la norme dominante de l'autostop. L'autostop véhiculait une mauvaise image dans les médias, à cause des drames qui pouvaient survenir (meurtres, vols, viols). Cette association a joué un rôle important, celui d'inventeur d'un dispositif, et d'innovateur en l'intégrant à un milieu social.

Une deuxième innovation d'ordre sémantique a également eu lieu. Le terme covoiturage est importé du Canada par l'association, et à la fin des années 1980 il remplace l'autostop organisé (Vincent, 2008, p. 90). La généralisation d'internet a provoqué une augmentation importante de sites dédiés au covoiturage. Ceci a eu comme conséquence l'augmentation du nombre de covoitureurs, mais également l'émiettement de l'offre de sites de mise en relation, et l'apparition d'une concurrence entre des entreprises spécialisées, des collectivités locales, des associations, et des particuliers (Certu, 2007, p.21). Cette concurrence et la méfiance entre les différents acteurs les a empêché de nouer des alliances (Callon, 1986, p. 185), qui auraient permis le regroupement des annonces et des bases de données au niveau national, et ainsi la mise en place d'un dispositif de mise en relation unique. Aujourd'hui, le site BlaBlaCar est leader sur le marché français du covoiturage, en progression constante, essentiellement orienté sur les trajets longues distances (350 km en moyenne selon son président-fondateur¹⁷).

Au niveau du covoiturage domicile-travail un certain nombre d'applications concurrentes sont développées par divers prestataires, mais elles peinent à se diffuser(a2).

I.3. La voiture partagée

Les moyens de transport à la base privés devenus publics (i.e. la voiture personnelle) ont donné naissance à des systèmes de transport à priori individuels mais devenus collectifs (e.g. services de covoiturage, autopartage ...). De tels systèmes ont vu le jour suite au développement de plusieurs travaux visant l'amélioration de la qualité de vie en considérant particulièrement le développement du domaine de transport dans un contexte mélioratif. Amélioration en termes des services fournis aussi bien sur le plan individuel que collectif et dont l'objectif premier consiste essentiellement à améliorer les conditions de vie des individus d'un côté et augmenter leur satisfaction de l'autre. Cependant, aussi évolués soient-ils, ces systèmes ne sont pas passibles d'efforts supplémentaires afin d'en optimiser le fonctionnement . Dans ce qui suit, nous passons en revue les principales études et approches s'inscrivant dans le contexte de la voiture partagée. Cette dernière a donné lieu à deux catégories selon le type de partage (Figure I.1)du véhicule :

- Autopartage : et il s'agit du partage d'un même véhicule par période de temps par des personnes différentes à chaque fois.
- Covoiturage : partage d'un véhicule pendant un même intervalle de temps par plusieurs personnes pour parcourir une partie ou la totalité d'un trajet commun.

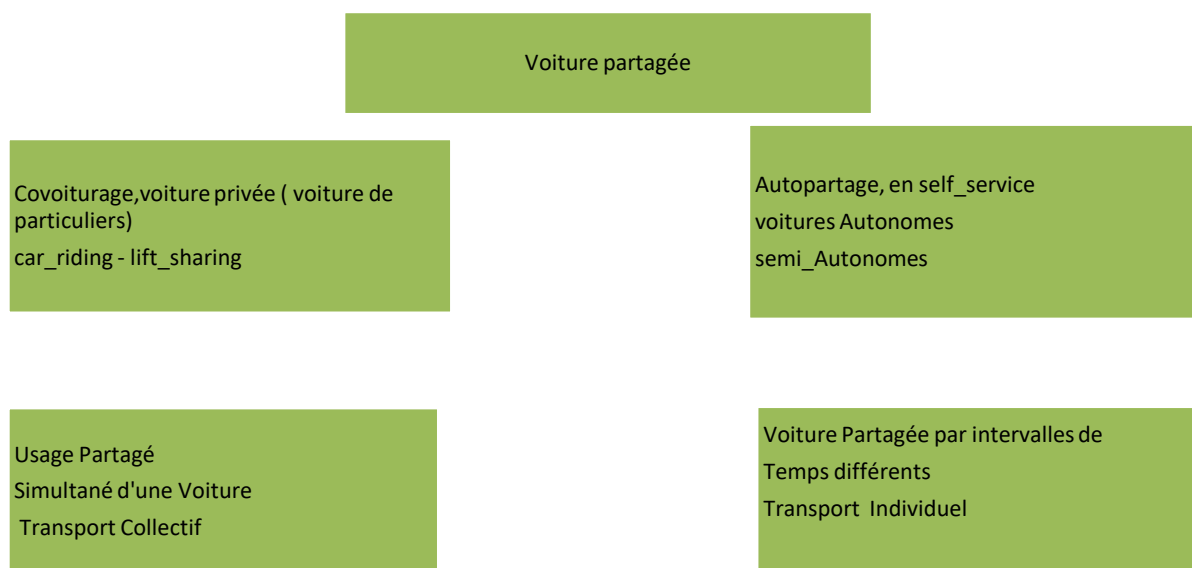


Figure I.1 : La voiture partagée : Autopartage ou Covoiturage ? (1)

I.4. Autopartage

Comme son nom l'indique, l'autopartage considère le partage d'un véhicule par une collectivité à des fins d'en optimiser l'usage et tente entre autre de solutionner la future pénurie mondiale de pétrole. Les chiffres et statistiques relevés en France (41) ne peuvent que le confirmer. Il s'agit en effet d'une forme de partage de véhicule particulier pour effectuer un trajet répondant à un besoin de déplacement bien déterminé. L'autopartage, ainsi

défini, est un service offert par une entreprise, une association ou une agence publique à toute personne en droit d'y accéder, mettant à sa disposition une flotte de véhicules en libre-service dont elle aura à partager l'usage avec d'autres membres(1).

I.5. Covoiturage

I. 5.1 . Définitions

Définition 1

Le covoiturage (dit carpooling, ride-sharing ou encore lift-sharing en anglais) se réfère à l'utilisation partagée d'une voiture par un conducteur non professionnel et un ou plusieurs passagers généralement pour effectuer des déplacements ensemble . Il existe deux façons d'utiliser le ou les véhicules.

- Utilisation à tou de rôle du véhicule de chacun des covoiturés, qui est ainsi alternativement conducteur et passager;
- Utilisation d'un seul véhicule et participation des passagers aux frais de déplacement (essence, péage).

Définition 2

Le covoiturage consiste en l'utilisation d'un véhicule de transport terrestre (le plus souvent une voiture) partagé par le conducteur avec 1 ou plusieurs passagers, dans le cadre d'un trajet effectué par le conducteur à son propre compte et qui ne donne lieu à aucune rétribution autre que celle du partage des frais (définition officielle au covoiturage à l' (article L3132-1) du code des transports(a4).

Définition 3

consiste à partager un véhicule avec des personnes qui effectuent le même trajet. Selon le ministère de l'écologie du développement durable et de l'énergie, en charge des transports, le covoiturage peut être défini comme « l'utilisation en commun d'un véhicule terrestre à moteur par un conducteur non professionnel (le covoitureur) et un ou des passagers majeurs (le/les covoiturés) sur un itinéraire ou section d'itinéraire commun ». C'est par exemple le cas pour des trajets réguliers domicile travail. Le covoitureur est généralement le propriétaire du véhicule, le trajet peut être réalisé à titre gracieux ou en échange d'une participation aux frais (péage, carburant). Il est précisé clairement que le transport des mineurs est soumis à une déclaration d'assurance.

Définition 4

Cette deuxième forme de partage de voiture particulière, pareille à sa précédente, connaît aujourd'hui un succès remarquable, grâce notamment aux avantages qu'elle a contribué à apporter. Elle a de fait soigné l'image environnementale, sociale et économique de la voiture réduisant les émissions de CO2, les dépendances à la voiture, les frais de transport engagés dans le cadre d'une mobilité individuelle, etc. Par ailleurs, outre les avantages économiques et écologiques, le covoiturage permet aussi de restaurer une certaine communication qui a disparu dans les transports en commun et qui a demeuré absente même avec le concept de l'autopartage. C'est ainsi que le covoiturage a permis, de par le regroupement de personnes se connaissant ou pas, de créer et de fortifier les liens sociaux.

Compte tenu de la montée en puissance du concept de la voiture partagée et plus particulièrement du covoiturage, mettant à profit les contributions qu'il est susceptible d'apporter à la société, nous avons choisi de diriger nos travaux vers ce contexte. Par la suite, nous proposons de nous approfondir dans ce sujet pour en détailler les axes principaux. Nous en fournissons alors une étude détaillée avec une liste non exhaustive des systèmes existants avant de définir la manière dont nous abordons le sujet de par notre approche ; laquelle se focalise essentiellement sur l'absence dans les travaux existants de certains concepts pourtant pertinents(1).



Figure I.2 : Partage du véhicule.

I. 5.2. Classification du problème de covoiturage

Selon les différentes procédures d'utilisation du service de mise en commun de voitures, nous classifions le problème de covoiturage sous deux formes différentes :

1- Problème de covoiturage quotidien (Daily Car Pooling Problem, DCPP) .

2- Problème de covoiturage à long terme (Long-term Car Pooling Problem, LTCPP).

DCPP

Dans le DCCP Un certain nombre d'utilisateurs déclarent leur disponibilité pour récupérer ou rapporter d'autres utilisateurs un jour donné. Par conséquent, ces utilisateurs sont considérés comme des serveurs, Et les autres utilisateurs récupérés ou rapporteurs sont considérés comme des clients. Ensuite, le problème consiste à attribuer des clients aux serveurs et à identifier les routes à parcourir par les serveurs. Puisque dans le DCCP, les serveurs et les clients sont connus à l'avance, L'objectif est de construire le chemin à partir de chaque serveur et de passer le plus grand nombre possible de clients tout en respectant la capacité de la voiture et les contraintes de la fenêtre de temps, et de minimiser le coût total du voyage Le modèle DCCP est basé sur l'horaire quotidien, de sorte que les participants changent tous les jours. C'est un modèle habituellement utilisé par le site Web commercial qui organise un service de covoiturage quotidien parmi les différents membres(a 3).

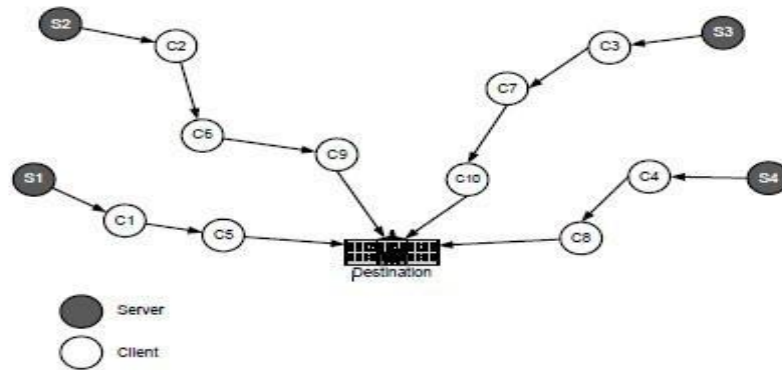


Figure I.3 : un exemple de DCCP(a 3).

LTCPP

Au contraire, dans le LTCPP, chaque utilisateur doit agir à la fois comme serveur et client et une solution consiste à définir des pools de voitures où chaque utilisateur, à son tour, à différents jours, ramènera les membres restants du groupe. L'objectif est de minimiser la quantité de véhicules utilisés et la distance totale parcourue par tous les utilisateurs, en respectant de la capacité de la voiture et des contraintes de la fenêtre de temps.

Le LTCPP peut être considéré comme une combinaison d'un problème de clustering et d'un problème de routage. Il faut que les membres du groupe soient relativement proches l'un de l'autre afin d'identifier l'itinéraire et le calendrier qui arrangent tout le groupe.

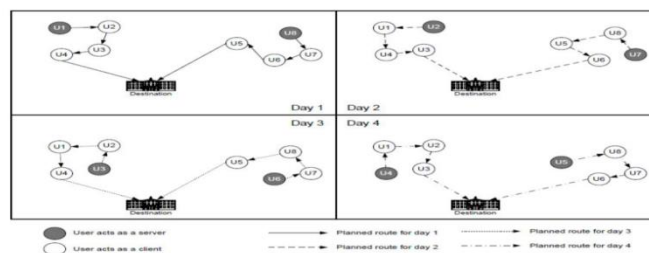


Figure I.4 : un exemple de LTCPP(a 3).

Le modèle LTCPP est un modèle de regroupement de voitures plus stable, les utilisateurs de LTCPP ne changeront pas fréquemment pendant une période relativement longue. Ce modèle est généralement utilisé par les grandes entreprises, les organisations et les universités qui fournissent un service de covoiturage pour leurs employés ou leurs étudiants.

Le LTCPP est un sujet de recherche plus précieux, car il a ses propres caractéristiques qui sont différentes des autres problèmes de routage des véhicules. Le problème nécessite de regrouper les utilisateurs dans des groupes en fonction d'un calendrier à long terme, et chaque membre du groupe doit agir en tant que serveur sur un autre jour.

I. 5.3 . TYPES DE DÉPLACEMENTS ET DE COVOITURAGE

I. 5.3.1. Covoiturage régulier

Ce type désigne les voyages et trajets qui sont d'une fréquence régulière. Il s'agit en effet des déplacements qui se font de manière hebdomadaire, c'est-à-dire au moins une fois par semaine. Le covoiturage régulier désigne aussi des voyages plus fréquents pour englober même ceux ayant lieu tous les jours (i.e. quotidiens). Nous faisons ici référence aux déplacements nécessaires pour rejoindre son domicile ou aller à son travail (i.e. domicile-travail, travail-domicile). De manière similaire, les trajets reliant le lieu de résidence à celui relatif aux études (i.e. trajets scolaires) s'inscrivent dans le contexte du covoiturage régulier. Il s'agit donc de trajets usuels prenant généralement effet à des dates et heures connues et concernent dans la plupart des cas des distances courtes. Bien avant que le covoiturage existe, ce concept de « partage de trajets » était bien développé entre le personnel d'une même entreprise, groupe d'étudiants fréquentant la même université ou école, etc.

I.5.3.2 . Covoiturage occasionnel

Ces trajets n'ont, comme leur nom l'indique, lieu que sil' occasion se présente; c'est-à-dire qu'ils n'ont pas (ou peu souvent) vocation à se répéter dans le temps. Il s'agit essentiellement de trajets de longue distance, généralement supérieure à 50 kilomètres. Nous pouvons en citer à titre d'exemple les déplacements ayant lieu aux départs des vacances, en périodes touristiques, en début et en fin de week-end. Le covoiturage occasionnel est la forme la plus commune et aussi la plus connue parmi les trois types qui existent. Ce type particulier concerne principalement les voyages entre les villes (ou pays) pour des trajets de longue distance planifiés bien à l'avance.

I. 5.3.3 .Covoiturage événementiel

Le covoiturage événementiel peut être considéré comme une variante du covoiturage de loisirs (voire aussi du covoiturage de longue distance), avec quelques particularités propres :
il s'effectue sur des distances relativement longues, pouvant atteindre l'échelle régionale, voire la dépasser (salon, grande manifestation sportive ou culturelle, fête d'envergure, etc.) ;
il est plutôt de nature exceptionnelle, même s'il peut présenter parfois une certaine récurrence (supporters d'une équipe sportive par exemple) ;
il peut se pratiquer entre personnes ne se connaissant pas à l'avance(1).

I. 6 .La différence entre l'autopartage et le covoiturage

L'autopartage est un service de transport offert dans certaines agglomérations qui, comme le covoiturage, fait partie des moyens de lutter contre l'explosion du parc automobile. Il est nécessaire de faire la distinction entre autopartage et covoiturage. Le covoiturage est un mode de transport qui consiste à partager l'usage d'un véhicule sur un même parcours au même moment (généralement pour un déplacement de type domicile – travail) ; il y a donc plus d'une personne dans le véhicule. Au contraire, l'autopartage consiste plutôt en un service de location

de voiture réservé aux abonnés qui leur permet en tout temps, d'accéder à une automobile sans en être le propriétaire. Il y a donc dans ce cas plusieurs usagers-conducteurs pour un même véhicule ; de plus il est possible qu'un véhicule d'autopartage ne soit occupé que par une seule personne. On constate que les compagnies qui proposent ce service sont généralement constituées d'une flotte de véhicules affectés dans différents stationnements répartis sur le territoire d'une agglomération. Ces véhicules peuvent être réservés à l'heure ou à la journée suivant les disponibilités [07].

I.7.Heuristiques

L'heuristique se réfère à des techniques basées sur l'expérience pour la résolution de problèmes, l'apprentissage et la découverte. Lorsqu'une recherche exhaustive n'est pas pratique, les méthodes heuristiques sont utilisées pour accélérer le processus de recherche d'une solution satisfaisante.

Une heuristique est conçue pour résoudre un problème qui ignore si la solution peut se révéler correcte, mais qui génère généralement une bonne solution ou résout un problème plus simple qui contient ou croise la solution du problème plus complexe. La Méthode heuristique peut accomplir sa tâche en utilisant des arborescences de recherche. Cependant, au lieu de générer toutes les branches possibles de la solution, une heuristique est sélective à chaque point de décision, et elle sélectionne les branches plus susceptibles de produire des résultats que les autres branches. Les études de l'algorithme basé sur les fonctions d'enregistrement, et l'approche basée sur la simulation entrent dans la catégorisation de l'heuristique(1).

I. 7.1. Saving Functions Based Algorithm(SFBA)

Dans l'Algorithme SFBA, une routine de traitement de l'information heuristique est conçue pour supporter une correspondance efficace dans les systèmes de covoiturage. Ceux-ci sont basés sur des fonctions d'épargne et appartiennent à deux macros classes distinctes d'algorithmes pour donner deux modèles différents de ce problème. Le travail est très axé sur la modélisation du problème au lieu de la phase de résolution. Les méthodes de résolution sont relativement simples en associant différents utilisateurs avec le support du modèle de carrosserie. L'approche est prouvée pour pouvoir fournir un pourcentage important pour économiser les distances de déplacement dans les applications réelles. Cependant, l'approche dépend fortement de la distribution des utilisateurs, seuls les repères avec les utilisateurs distribués en cluster ont pu obtenir de bons résultats(a 3).

I. 7.2. The simulation-based method

La méthode basée sur la simulation utilise une division et une conquête approche. L'heuristique utilisée dans l'étape de division est l'algorithme de clustering K-means qui permet de classer les objets en fonction des attributs en plusieurs groupes. Le regroupement se fait en minimisant la somme des carrés de distances entre les utilisateurs et le centre de concentration correspondant. Les auteurs estiment que la proximité géographique ne garantit pas une bonne correspondance entre les utilisateurs, de sorte que l'heure de départ et d'arrivée des utilisateurs a également été considérée comme faisant partie de la distance entre les utilisateurs et le centre de concentration correspondant. Le processus commence par utiliser l'algorithme de clustering K-

means pour diviser tous les utilisateurs en petits clusters de sorte que chaque petit cluster puisse être traité par le logiciel d'optimisation dans un délai acceptable .

I. 8. Métaheuristiques

La métaheuristique désigne une méthode de calcul qui optimise un problème en essayant itérativement d'améliorer une solution candidate en fonction d'une mesure de qualité donnée. Les métaheuristiques font peu ou pas d'hypothèses sur le problème optimisé et peuvent rechercher des espaces très étendus de solutions candidates. Ils peuvent accepter temporairement des solutions pires lors de la procédure d'optimisation. Ainsi, il est possible de conduire la recherche de l'optimum local. Les règles pour accepter les pires solutions sont appelées diversifications.

Avec le mécanisme de diversification, les métaheuristiques peuvent générer un optimum global et sont insensibles aux solutions initiales. Cependant, les métaheuristiques ne garantissent pas une solution optimale. En outre, les métaheuristiques contiennent de nombreux paramètres empiriques sensibles à la casse, ce qui peut entraîner des difficultés pour les implémentations pratiques .

I. 8. 1. Trajectory-based Metaheuristics (TBM)

L'algorithme basé sur la trajectoire opère généralement sur une solution à la fois, ce qui trace un chemin dans l'espace de recherche à mesure que les itérations se poursuivent. Les chemins sont exécutés par des procédures itératives qui permettent de passer d'une solution à une autre dans l'espace de la solution. Les parcours partent d'une solution générée aléatoirement ou obtenue à partir d'un autre algorithme d'optimisation, appelé solution initiale.

A chaque itération, la solution actuelle est remplacée par une autre sélectionnée dans l'ensemble des candidats voisins. Un meilleur déplacement ou une meilleure solution est toujours accepté, alors qu'un déplacement moins bon peut être accepté avec une certaine probabilité, illustrée à la figure I.5. Les étapes ou les déplacements suivent une trajectoire dans l'espace de recherche, avec une probabilité non nulle que cette trajectoire atteigne l'optimum global.

Le processus de recherche est arrêté lorsqu'une condition donnée est remplie, par exemple un nombre maximal de générations, la recherche d'une solution avec une qualité cible ou l'absence d'améliorations pour un temps donné, etc.

Ce type de métaheuristique effectue les mouvements au voisinage de la solution actuelle , ils ont donc un caractère perturbatif .

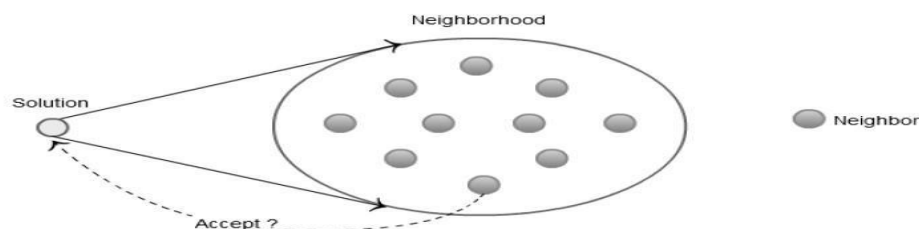


Figure I. 5 : principes de la métaheuristique basée sur la trajectoire(a 3) .

I. 9. Conclusion

Il est vrai que les systèmes de partage de véhicules aident à l'amélioration des conditions environnementales dans lesquelles évoluent les individus. Le concept de covoiturage présente particulièrement de multiples avantages, dont la réduction du nombre de voitures en circulation au kilomètre. En effet, la voiture personnelle est devenue, grâce à ce concept, un moyen de déplacement collectif accessible au grand public. Ces systèmes représentent en ce sens un moyen efficace pour minimiser le taux d'émissions de CO₂ et gaz à effet de serre aussi bien que pour réduire les budgets alloués au transport.

Le covoiturage apporte la réponse adéquate à des interrogations longtemps restées en suspens. C'est dans ce cadre que nous avons dirigé nos travaux pour proposer une approche innovante dont les fondements ont été construits sur la base d'une étude approfondie de l'histoire du covoiturage et des principes et concepts faisant défaut aux systèmes érigés de par le développement de ce phénomène.

Notre système, baptisé CODAC pour Covoiturage Optimisé Dynamique basé sur les Agents communicants, intègre comme son nom l'indique le concept d'optimisation dans une alliance avec les Systèmes Multi-Agent (SMA). Cette alliance vise la mise en place de traitements efficaces dans un système performant fournissant une qualité de service optimale. Par ailleurs, traiter une problématique d'optimisation dans un contexte d'affectation en temps réel engendre une complexité d'ordre exponentiel qui fait entrave à la conduite à bon escient d'un système doté d'une performance optimale sur tous les niveaux. Par conséquent, nous ne pouvons prétendre à l'optimalité dans tous les sens et sous tous les points de vue mais cherchons les meilleurs compromis possibles : évitant les situations conflictuelles, s'exécutant dans des délais de réponse raisonnable, fournissant des réponses optimisées, etc. Pour ce faire, nous sommes allés puiser dans les théories des graphes, les méthodologies de résolution basées sur l'intelligence artificielle distribuée, les méthodes et algorithmes de recherche opérationnelle, les algorithmes d'optimisation multi-objectif, les SMA, etc. Tous ces concepts ont été consacrés et utilisés dans un seul sens où nous avons dirigé nos travaux. Dans cette perspective, nous visons en effet de mettre en place un processus d'affectation dynamique optimisé décentralisé basé sur la subdivision du réseau de desserte. Une mesure de performance efficace pour une qualité de service optimisée et un système compétitif prime sur tout autre objectif dans la vision où nous menons nos travaux. Toutes ces notions ainsi que la manière dont elles opèrent pour la bonne marche de notre système seront détaillées dans le chapitre suivant.

II. CHAPITRE II

L'optimisation Et Les Metaheuristiques

II.1. Introduction

La résolution de problèmes d'optimisation est devenue un sujet central en recherche opérationnelle, le nombre de problèmes d'aide à la décision pouvant être formalisés sous la forme d'un problème d'optimisation étant en forte croissance. En outre, l'homme cherche à améliorer sa vie quotidienne, l'homme aime la perfection et sans qu'il se rend compte, il essaye à minimiser ses charges, son loyer ou la consommation de sa voiture. Il tente toujours à optimiser : que ce soit minimiser ses dépenses ou maximiser ses biens. De nos jours, l'optimisation est devenue un domaine indispensable pour résoudre plusieurs problèmes que se soit dans l'industrie ou d'autres secteurs. En effet, nous avons assisté ces dernières années à une croissance très rapide des travaux utilisant les méthodes d'optimisation. Cette tendance peut être observée dans tous les domaines de la science. Nous aborderons dans ce chapitre les définitions générales des méthodes d'optimisation qui se divisent sous deux volets déterministes et non déterministes.

II.2. Optimisation

II.2.1. Optimisation : Concept et définition

L'optimisation a été introduite dans un souci d'amélioration des services fournis, peu importe le domaine auquel ils s'appliquent. Un problème d'optimisation concerne l'exécution de méthodes spécifiques en quête d'un **optimum**. Ce dernier peut être une valeur maximisant ou minimisant une fonction **f**, dite **fonction objectif** ou **fonction de coût** ; elle est encore appelée **critère d'optimisation**.

II.2.2. Délimitation de l'espace de recherche

L'espace de recherche où s'appliquent les méthodes d'optimisation pour l'extraction d'une solution optimale est restreint et délimité grâce aux limites imposées par les contraintes d'optimisation. Nous distinguons deux types de contraintes d'inégalité :

Celles de type $B_{\text{inf}} \leq x_i \leq B_{\text{sup}}$

Celles de type $(x_c) \leq 0$ ou $(x_c) \geq 0$

II.2.3. Différents critères d'optimalité pour différents types d'optimum

D'*optimal* à *optimisé* il existe une différence se référant au degré d'optimalité d'une valeur de **f** pour un point choisi dans un espace de recherche global et l'étendue de cette optimalité par rapport aux différentes valeurs de **x** telles que définies dans la précédente section :

- Un **maximum global**

-Un **maximum local fort**

-Un **maximum local faible**

II.2.4. Classification des problèmes d'optimisation

Tableau II.1. Classification des problèmes d'optimisation (19)

Caractéristiques du problème		Type du problème	
<i>Variables de Décision</i>	<i>Nombre</i>	<i>1</i>	<i>Monovariante</i>
		<i>>1</i>	<i>Multivariable</i>
	<i>Type</i>	<i>Nombre réel continu</i>	<i>Continu</i>
		<i>Nombre entier</i>	<i>Entier ou discret</i>
		<i>Permutation sur un ensemble fini de nombres</i>	<i>Combinatoire</i>
<i>Fonction Objectif</i>	<i>Type</i>	<i>Fonction linéaire des variables de décision</i>	<i>Linéaire</i>
		<i>Fonction quadratique des variables de décision</i>	<i>Quadratique</i>
		<i>Fonction non linéaire des variables de décision</i>	<i>Non linéaire</i>
<i>Formulation du problème</i>	<i>Type</i>	<i>Avec des contraintes</i>	<i>Contraint</i>
		<i>Sans contraintes</i>	<i>Non contraint</i>

Figure II.1 : Classification des problèmes d'optimisation (19) .

II.2.5. Optimisation mono-objectif

Implique le fait de considérer un seul critère à optimiser. Il s'agit ici d'une catégorie de problèmes d'optimisation relativement « faciles » à résoudre. Le terme *facile* est ici utilisé non pas pour désigner le degré de difficulté du problème mais par relativité aux problèmes multiobjectifs puisque ne présentant pas de conflit d'intérêt à privilégier l'un ou l'autre des critères d'optimisation comme c'est le cas quand une multitude de critères doivent être considérés(56).

II.2.6. Optimisation multi-objectif

L'optimisation multi-objectif s'incruste dans la considération d'une multitude de critères à prendre en compte en même temps sachant qu'ils peuvent à fortiori être contradictoires et engendrer des conflits d'intérêts. Cette extension est d'autant plus nécessaire que de plus en plus de problèmes nécessitent la considération de manière simultanée d'une multitude d'objectifs. L'optimisation multicritère consiste donc à choisir parmi un ensemble infini d'alternatives une seule (ou plusieurs) lorsqu'une multitude de critères doivent être considérés . L'ensemble de ces alternatives varie généralement dans un domaine continu. Depuis plus de quatre décennies déjà, la problématique multicritère n'a pas arrêté d'évoluer témoignant de la naissance et du développement d'une grande panoplie de méthodes. Deux groupes distincts ont émergé pour réaliser la classification de celles-ci; le premier est constitué des méthodes qui s'appuient sur un critère unique de synthèse, lesquelles sont dites *classiques* ; alors que le deuxième inclue des méthodes réactives au sens où elles intègrent un processus interactif de décision(57)(58) .

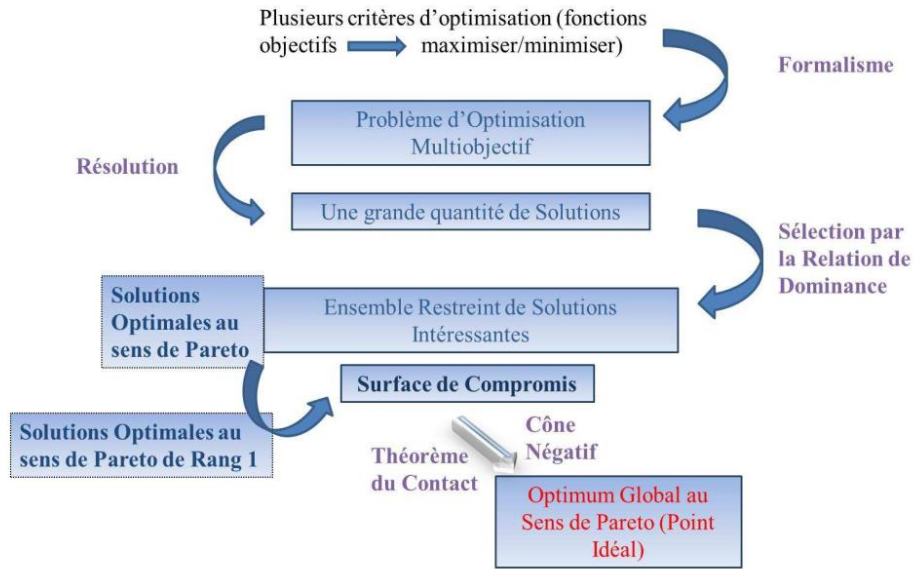


Figure II.2 : Optimisation multi-objectif et optimalité au sens de Pareto.

II.2.7. Les méthodes d'optimisation

Selon le cas, qu'il s'agisse d'un problème monovarié ou multivarié, continu ou discret, etc. une méthode d'optimisation adéquate est choisie avec soin afin de bien cadrer avec le contexte et résoudre le problème de manière efficace. Entre méthodes exactes, métaheuristiques, hybrides ou autres, chercheurs et praticiens ont à leur disposition un large panel de choix de méthodes d'optimisation qu'ils peuvent adopter(56).

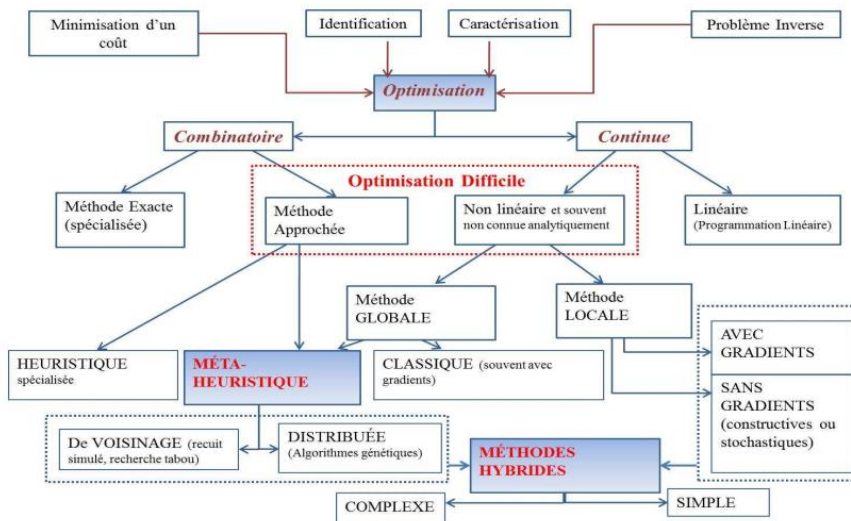


Figure II.3 : Méthodes d'optimisation monobjectif (56).

Les méthodes scalaires

Elles regroupent un ensemble de méthodes dont l'approche pour la résolution d'un problème d'optimisation multiobjectif est la plus évidente. Cette approche appelée « *approche naïve* » a pour but de reformuler le problème d'optimisation de façon à revenir à une optimisation monobjectif. La manière la plus intuitive de par laquelle ce procédé est réalisé est l'**agrégation** en utilisant la sommation de l'ensemble des fonctions objectifs en une seule

Les méthodes interactives

Les méthodes regroupées sous cette classe forment la famille des méthodes progressives et permettent de chercher une solution unique. Elles se basent pour cela sur une interactivité avec l'utilisateur qui a l'occasion de choisir au fur et à mesure de l'exécution de la méthode la solution qui correspond le mieux à ses préférences par rapport au compromis qu'il souhaiterait voir effectué sur les différentes fonctions objectifs.

Les méthodes floues

Ces méthodes sont inspirées de la logique floue où la logique binaire (vrai ou faux) est délaissée en faveur de l'introduction d'un certain degré de *flexibilité* ou *élasticité* en vue de pouvoir tolérer les incertitude et imprécision des connaissances humaines. La méthode de *Sakawa* ou encore la méthode de *Reardon* sont des exemples à citer dans cette catégorie.

Les méthodes basées sur une métaheuristique

Ces méthodes, apparues depuis les années 1980, comprennent la méthode du *recuit simulé*, les *algorithmes génétiques*, la méthode de *recherche tabou*, les algorithmes de *colonies de fourmis*, etc. Elles ont pour objectif principal et commun entre elles, de résoudre au mieux les problèmes d'optimisation difficile.

A. Le recuit simulé (Simulated Annealing)

La méthode du *recuit simulé* est inspirée du procédé du *recuit* et a pour principe de partir d'une configuration donnée (e.g. placement initial de tous les composants d'un circuit électronique, configurations de basse énergie de matériaux magnétiques désordonnés) pour aboutir au résultat escompté. Ce résultat concerne l'optimum global, ou un résultat qui s'en rapproche, atteint via l'application de transformations élémentaires en un nombre fini d'itérations.

B. La recherche tabou (Tabu Search)

La méthode de recherche avec tabous, formalisée en 1986 par F. Glover dans [61], est encore appelée recherche tabou ou simplement méthode tabou (Tabu Search, TS). L'intérêt principal de cette méthode tient de la mise à profit de techniques et mécanismes inspirés de la mémoire humaine. En effet, contrairement à la méthode du recuit simulé, précédemment définie et qui est totalement dépourvue de mémoire, la recherche tabou garde en mémoire les traces des différentes étapes de son application [62] [59].

C. Les algorithmes Évolutionnaires (Evolutionary Algorithms)

Les algorithmes évolutionnaires (AE) sont apparus à la fin des années 1950. Ils utilisent des techniques de recherche inspirées par l'évolution biologique des espèces. De premier abord, les algorithmes évolutionnaires ont suscité un intérêt minime, ceci étant principalement dû à leur important coût d'exécution. Cependant, l'intérêt porté à ces techniques a considérablement crû durant les deux dernières décennies, et ce grâce à l'augmentation vertigineuse de la performance des calculateurs, aujourd'hui très puissants [63].

D. Les algorithmes de colonies de fourmis (Ant colony)

Énoncée par Coloni, Dorigo et Maniezzo dans [64], la méthode de colonies de fourmis se base sur le comportement des colonies de fourmis et s'efforce donc de simuler la capacité collective observée chez ces dernières pour la résolution de certains problèmes. Notons que les différents membres des colonies de fourmis sont par ailleurs dotés de facultés très limitées. Plusieurs recherches se sont orientées vers l'étude du mode de survie et des habitudes comportementales des fourmis qui sont comptés comme étant l'une des espèces les plus prospères [64].

Les algorithmes de colonies de fourmis sont dotés de plusieurs caractéristiques intéressantes dont nous pouvons citer essentiellement :

- Le parallélisme intrinsèque élevé,
- La flexibilité : une colonie de fourmis est dotée d'un fort degré de flexibilité permettant son adaptation à des modifications de l'environnement,
- La robustesse : une colonie est apte à maintenir son activité si quelques-uns des individus la formant sont défaillants,
- La décentralisation : cette caractéristique met encore une fois l'accent sur l'aspect réparti (i.e. distribué) de la démarche puisqu'une colonie n'obéit pas à une autorité principale et centralisée,
- L'auto-organisation : une colonie est dotée d'un certain degré d'autonomie, elle est ainsi capable de trouver d'elle-même une solution qui n'est pas connue à l'avance. [56].

II.2.8. Les méthodes d'aide à la décision

Il s'agit de méthodes d'optimisation qui permettent de répondre à plusieurs problématiques mais dont une particularité essentielle est celle de ne pouvoir travailler sur des ensembles de valeurs continues. Contrairement à la relation de dominance précédemment définie, ces méthodes permettent d'extraire un ensemble de solutions liées par une relation d'ordre : *partiel* s'il s'agit d'un ensemble de solutions ou *total* si une et une seule solution est extraite(21)(23)(19).

II.3. Les Algorithmes Métaheuristiques Bio-Inspirés

Aujourd'hui, un ahurissant succès est mérité grâce à la modélisation de l'intelligence biologique aboutissant en ce que nous appelons «les algorithmes méta-heuristiques bioinspirés» qui ont démontré leurs forces face aux différents problèmes complexes en cherchant toujours à trouver la solution optimale à partir d'un nombre fini de solutions existantes.

Ces algorithmes comprennent les réseaux de neurones artificiels, les calculs évolutionnaires, l'intelligence collective, les systèmes immunitaires artificiels et les systèmes d'organes humains. Ils fournissent des résultats de haute performance dans les études expérimentales, et peuvent souvent être appliqués sans beaucoup de connaissances sur le problème, ce qui les rend très appropriés pour diverses circonstances. En contrepartie, il est souvent difficile de comprendre pourquoi ils sont performants dans un contexte particulier(24).

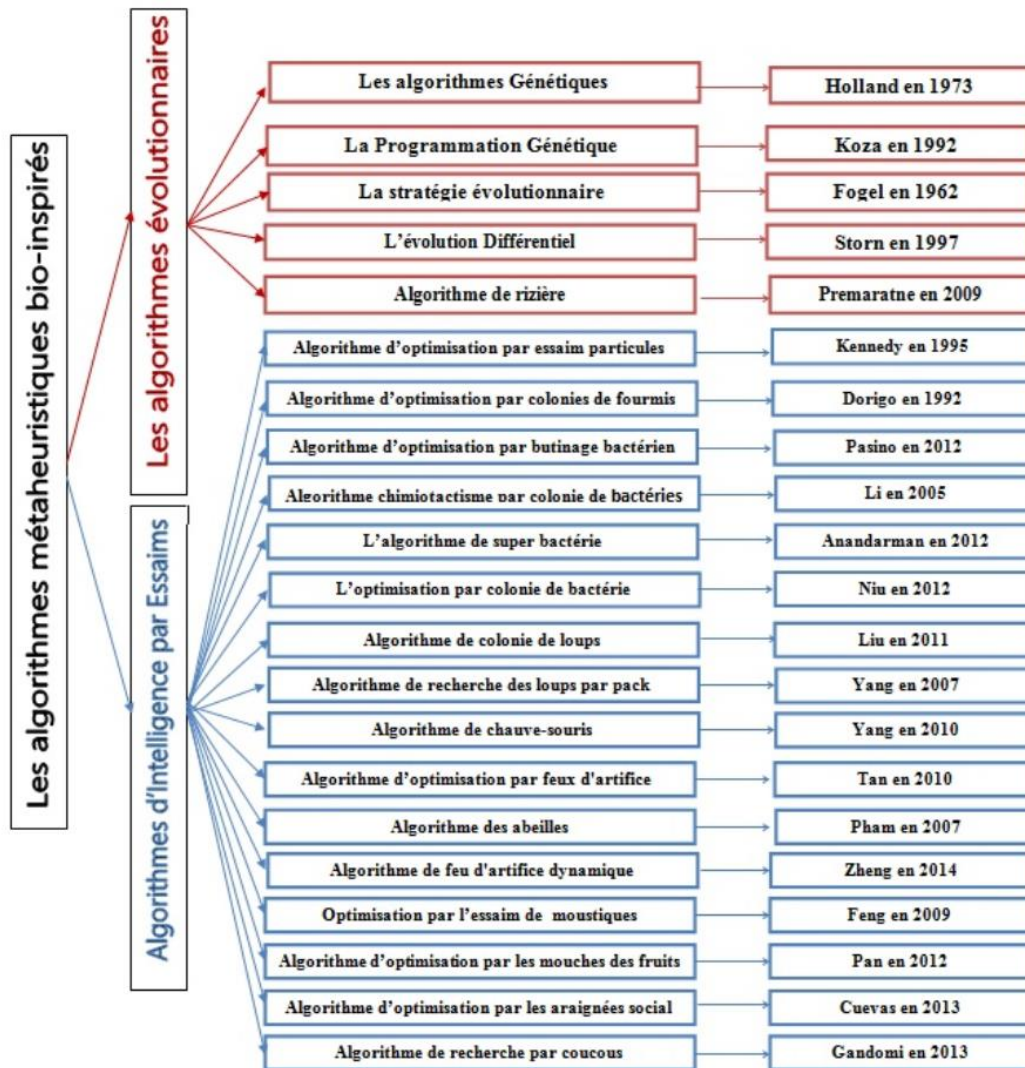


Figure II.4 : Taxonomie des algorithmes méta-heuristiques bio-inspirés.

II.3.1. Le bio-mimétisme

La nature est une source puissante d'inspiration pour les prospecteurs puisqu'elle fournit des phénomènes extrêmement divers, dynamiques, robustes, complexes et fascinants. Elle trouve toujours une solution pour résoudre ses problèmes, et maintenir l'équilibre parfait entre ses composants. Les êtres vivants ont réussi à survivre sur la terre au cours des quatre derniers milliards d'années. La principale raison pour un tel succès est certainement leur capacité de s'adapter à l'évolution des environnements défavorables. Ils possèdent étonnantes facultés à apprendre à partir des situations inconnues, d'adapter leurs comportements et leurs formes aux changements environnementaux, à l'autoreproduction, et à l'auto-réparation. Dans les décennies

passées, de nombreux efforts de recherches ont été concentrés dans ce secteur et une nouvelle ère est ouverte avec les algorithmes inspirés de la nature (bio-inspirés).

La bio-inspiration est l'imitation d'un processus biologique comme les métaphores ou les phénomènes naturels pour le développement de nouveaux algorithmes. Ce champ d'étude a tricoté vaguement un ensemble des sous-domaines liés aux thèmes du connexionnisme, le comportement social et l'émergence. Il est souvent étroitement lié au domaine de l'intelligence artificielle, comme beaucoup de ses activités sont liés à l'apprentissage artificielle. Les algorithmes bio-inspirés sont avérés significativement plus robustes et adaptables que les algorithmes traditionnels. Les notions de robustesse, d'émergence, d'auto-organisation d'adaptabilité, réactivité et de distribution sont donc sous-jacentes dans ces algorithmes et font même partie de leurs fondements. La première étape dans la construction d'un algorithme bio-inspiré est de construire des composants imitant le comportement de leurs homologues biologiques. Ces composants essaient ensuite d'atteindre l'objectif global défini pour eux(24).

II.4. Optimisation dans les problèmes de covoiturage

De manière spécifique à notre domaine d'intérêt, nous considérons ici les travaux relatés dans la littérature et ayant considéré un volet optimisation dans des problèmes de covoiturage. L'auteur d'une étude sur les taxis collectifs introduit des algorithmes d'optimisation dans ce contexte en application au problème de correspondance (matching) entre les trajets requis dans un contexte de voyages quotidiens de type domicile-travail (plusieurs-à-un) et travail-domicile (un-à-plusieurs). Aussi dans ce cadre I.H. Dridi a introduit de par ces travaux l'optimisation heuristique au problème du m-PDPTW (Pickup and Delivery Problem with Time Windows à plusieurs véhicules) prenant en compte le volet statique aussi bien que dynamique adoptant pour cela les algorithmes génétiques. Nous citons particulièrement ces travaux, puisque le PDPTW, variante du problème de VRP se rapproche considérablement du problème de covoiturage. En effet, en plus de l'existence des contraintes sur le temps et la capacité des véhicules, le PDPTW implique un ensemble de clients (piétons dans notre cas) et un ensemble de fournisseurs géographiquement localisés (pouvant être assimilés aux offres de covoiturage) et une correspondance adéquate doit être effectuée entre les deux(75)(76).

II.5.1 Complexité du problème de covoiturage dynamique optimisé

En matière d'optimisation difficile, deux types de problèmes d'optimisation sont discernés ; le premier est relatif à ceux dits *combinatoires* (*problèmes discrets*) et le deuxième à ceux dont les variables sont *continues*. La différenciation entre les problèmes discrets et continus a été considérée pour cerner le domaine de l'optimisation difficile. En effet, n'ayant pas de définition précise et stricte, deux types de problèmes distincts lui ont valu cette description :

- Le premier concerne les problèmes d'optimisation combinatoire pour lesquels il n'existe pas d'algorithmes *exacts rapides*(77).
- Le deuxième type est relatif aux problèmes d'optimisation dont les variables sont de type continu(1).

II.5.2. Avis et mesures

Se heurter à un problème de complexité importante peut fort probablement conduire à un échec du système si cette problématique n'est pas traitée. En effet, ayant des répercussions très significatives et accablantes, une complexité exponentielle induit la plupart du temps à une lenteur généralement importante dans les délais d'exécution du système. Par conséquent, afin de demeurer à l'échelle de la compétitivité et offrir un système riche en fonctionnalités tout en étant techniquement performant, nous nous intéressons à une nouvelle stratégie de résolution efficace et efficiente(1).

II.6. Les systèmes multi-agents au service de l'optimisation

Les systèmes multi-agents ont connu, depuis leur apparition aux états unis en 1970, un essor important au niveau mondial qui n'a pas encore été démenti. Faisant intervenir plusieurs entités hétérogènes pour la résolution d'un même et unique problème, et capables de communiquer de surcroît, les SMA ont changé la perception « *classique* » des méthodologies de résolution. Ce concept a par ailleurs l'avantage supplémentaire de coïncider avec une montée en puissance des nouvelles Sciences et Technologies de l'Information et de la Communication (STIC) changeant ainsi la vision que l'on a habituellement des interactions entre différentes entités intelligentes.

Développés dans le cadre d'applications sophistiquées, les SMA favorisent la mise en place de processus distribués sur différentes entités autonomes (ou semi-autonomes) pour en garantir l'optimalité(1).

II.6.1. SMA : le concept

Un SMA est un système composé d'un ensemble d'entités interactives et autonomes, appelées « *agents* ». Un concept étroitement lié aux SMA est la notion d'organisation décrivant la manière suivant laquelle les différentes entités opèrent et interagissent entre elles au sein d'un environnement spécifique.

Cette organisation décrit par ailleurs les agissements et communications entre ces entités régissant de ce fait les relations ainsi que la dynamique des interactions entre celles-ci(18).

Plus concrètement, un SMA est composé d'un environnement, d'un ensemble d'objets passifs (créés, perçus, modifiés et détruits par les agents), d'un ensemble d'agents actifs et

avec, les relations qui les relient entre eux, leurs opérations et compétences ainsi qu'un ensemble de lois universelles (81).

II.6.1.1. Définitions élémentaires

Plusieurs travaux se sont intéressés au concept des SMA et ont essayé de donner une définition précise à la notion d'*agent*. Nous tirons nos définitions ici d'études intégrant les SMA dans le domaine du transport distribué, notamment ceux évoqués dans l'état de l'art que nous avons réservé pour l'étude de l'existant dans le domaine du transport multimodal et comodal, etc(2)(82)(83).

Ainsi le terme agent a bénéficié de plusieurs identifications et spécificités qui se rapprochent incontestablement. Par ailleurs, la définition la plus récurrente est celle qui définit un **agent** comme étant une entité virtuelle (logicielle) ou physique autonome (84):

- capable d'agir dans un environnement
- dotée de capacité de communication la reliant directement aux autres agents
- mue par un ensemble de tendances (sous forme d'objectifs individuels ou d'une fonction de satisfaction voire même de survie qu'elle veut optimiser)
- en possession de ressources qui lui sont propres
- dotée d'aptitudes de perception de parties limitées de son environnement
- possédant une représentation partielle, voire même nulle de l'environnement dans lequel elle évolue
- dotée d'un certain nombre de compétences et apte à offrir un ensemble de services
- capable éventuellement de se reproduire
- suivant un comportement qui lui est spécifique dans le but d'atteindre ses objectifs fixés et ce en prenant en considération ses perceptions et représentations et en fonction aussi des communications et messages dont elle est réceptive. Ceci est faisable notamment grâce aux compétences et ressources dont elle est en possession.

Le cycle de vie d'un agent en est ainsi perçu et déduit directement à partir de cette définition. Trois grandes étapes permettent de retracer ainsi l'activité d'un agent dès sa création :

a. La **perception** de l'environnement : pour ce faire, l'agent use de ses connaissances et comportements

b. La **prise de décision** : appelée encore **délibération** consiste à décider de l'action suivante à entreprendre. Les connaissances et l'intelligence de l'agent aidant, cette fonctionnalité constitue une des spécificités les plus importantes faisant la particularité du concept agent. Elle symbolise en effet l'**autonomie** d'un agent.

c. **L'exécution** : consiste concrètement à effectuer l'action choisie lors de la précédente étape du cycle de vie de l'agent.

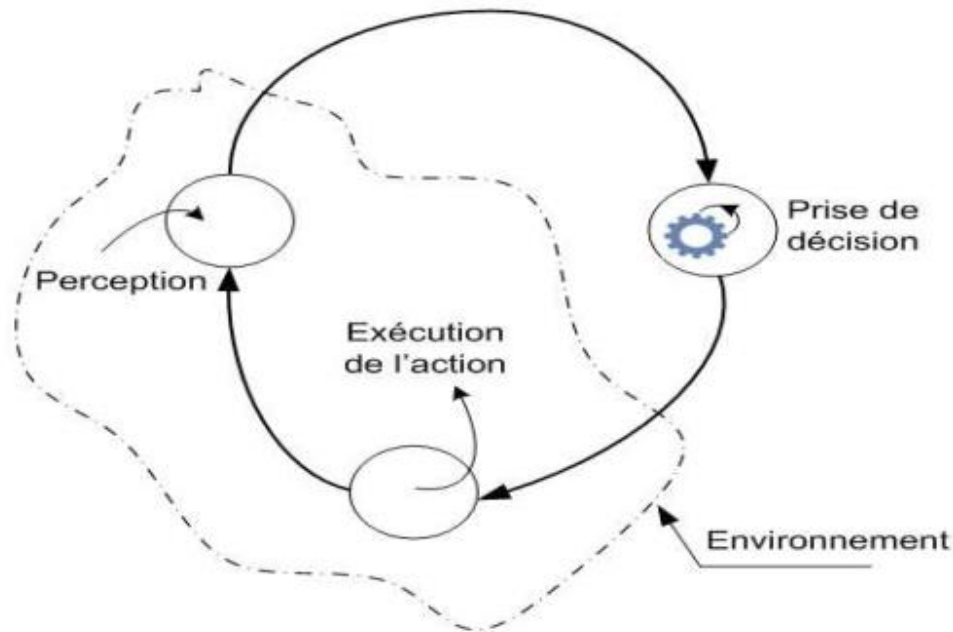


Figure II.5 : Cycle de vie d'un agent (2) .

II.6.1.2. Différents types d'agents

La différence dans la typologie des agents relève essentiellement de la *différence comportementale* par rapport à la manière et la rapidité d'exécution des actions qui leur sont spécifiques . Deux types principaux d'agents sont distingués dans la littérature et ont fait l'objet de plusieurs travaux (85)(86)(87) :

- Les **agents réactifs** : très rapides, leurs actions sont assimilées à des réflexes.
- Les **agents cognitifs** : contrairement aux précédents, ces agents possèdent un comportement réfléchi basé sur leurs connaissances de l'environnement, des autres agents et d'eux-mêmes(88).

II.6.1.3. Propriétés d'un agent

Tout agent correspondant à la définition précédente possède un certain nombre de propriétés lui faisant valoir le succès qu'il a reçu, ceci grâce à l'efficacité dont ce concept a fait preuve. Nous distinguons principalement cinq propriétés typiques des agents :

a. L'**autonomie** : se définit comme une existence indépendante : un agent autonome est, en effet, un agent dont l'existence ne dépend pas de celle des autres(89)(90) .

b. La **réactivité** : elle concerne l'aptitude d'un agent à réagir aux modifications survenant dans l'environnement dans lequel il évolue et d'adapter son fonctionnement et comportement (actions et ordre dans lequel elles sont entreprises) en fonction des changements perçus.

c. La **pro-activité** : tout agent est doté de certaines capacités dont une principale est son aptitude à agir de sa propre initiative pour essayer d'atteindre les buts et objectifs qu'il s'est fixé.

d. La **continuité** : concerne la capacité d'un agent à demeurer en continuelle activité sans pour autant être provoqué par un stimulus externe(91).

e. La **sociabilité** : la notion de communication intervenant dans les SMA est le principal moteur à un comportement social évolué.

II.6.2. Interaction et communication

Les interactions, telles que définies par Ferber dans , correspondent à une mise en relation directe et dynamique de deux agents ou plus, et ce par le biais d'un ensemble d'actions réciproques. Telles que énoncées précédemment, ces interactions sont un concept d'ordre d'importance très grand dans les organisations et constituent un concept inhérent et indissociable de celui des SMA(84)(94).



Figure II.6 : Les interactions sous leurs différentes formes.

II.6.3. Les techniques multi-agents intégrées au concept de covoiturage

Grâce à ses multiples contributions faisant la réussite d'une multitude de systèmes pour la résolution de problèmes complexes, plusieurs chercheurs ont ainsi concentré leurs efforts sur ce concept, non pas nouveau mais très innovant, et bénéficiant par la même occasion de l'apport considérable des nouvelles technologies de l'information et de la communication. La problématique de covoiturage n'a pas été épargnée non plus, donnant lieu à l'incorporation de ce concept dans quelques systèmes de covoiturage . Dans ce cadre favorable à l'efficacité et la rapidité de la communication (entre les utilisateurs et le système par exemple), chercheurs et praticiens ont ainsi cherché à tirer profit de l'évolution remarquable que connaît le domaine des réseaux et technologies de l'information et de la communication. Ceci a été enregistré dans un cadre révolutionnaire du covoiturage qui était resté, jusqu'ici, au stade de l'application pratique, sous forme de support en ligne pour la dépose et consultation d'offres et demandes(103)(104).

II.7.1. Optimisation dans le covoiturage dynamique

Considérant un seul critère ou plusieurs, l'optimisation ne peut qu'être bénéfique au sens des usagers du service de covoiturage. Dans le domaine du transport en général, les systèmes les plus réussis sont ceux qui offrent une assistance au voyageur dans ses déplacements tout en lui assurant de trouver les meilleures offres qui lui correspondent. Aujourd'hui, même les opérateurs de transport publique sont en quête de performances par rapport à cet aspect d'optimisation afin d'offrir une qualité de service susceptible d'attirer les individus. Dans cette quête des performances, des méthodes d'optimisation ont été consacrées implémentant des algorithmes jugés adéquats dans le cadre des situations auxquelles ils sont confrontés(1).

II.8. Les algorithmes génétiques

Les algorithmes génétiques (AG) appartiennent à la famille des algorithmes évolutionnistes (un sous-ensemble des méta heuristiques). Leur but est d'obtenir une solution approchée, en un temps correct, à un problème d'optimisation, lorsqu'il n'existe pas (ou qu'on ne connaît pas) de méthode exacte pour le résoudre en un temps raisonnable. Les algorithmes génétiques utilisent la notion de sélection naturelle développée au XIX^{ème} siècle par le scientifique Darwin et l'appliquent à une population de solutions potentielles au problème donné.

Les algorithmes génétiques ont la particularité de s'inspirer de l'évolution des espèces dans leur cadre naturel. Les espèces s'adaptent à leur cadre de vie qui peut évoluer, les individus de

chaque espèce se reproduisent, créant ainsi de nouveaux individus, certains subissent des modifications de leur ADN, certains disparaissent.... Un algorithme génétique va reproduire ce modèle d'évolution dans le but de trouver des solutions pour un problème donné(11) .

Il est nécessaire de définir quelques termes de base rencontrés utilisés dans les algorithmes génétiques (14)(16):

- **Individu** : solution potentielle du problème.
- **Chromosome (Génotype)** : solution potentielle du problème sous une forme codée (Chaîne de bits).
- **Gène** : partie élémentaire (Un bit dans une chaîne de bits) non divisible d'un chromosome.
- **Population** : ensemble fini d'individus (Solutions).
- **Progéniture** : est l'ensemble des nouveaux chromosomes obtenus par les processus de croisements et mutations.
- **Une génération** : n est l'ensemble des chromosomes obtenus lors de la *nième* itération.

Les algorithmes génétiques ont pour objectif d'explorer une partie de l'ensemble des solutions possibles afin de déterminer une solution la plus adaptée (i.e. une solution obtenant le meilleur score à une fonction objectif). Pour ce faire, ils fonctionnent à partir d'une population initiale de chromosomes, eux-mêmes composés de gènes.

Afin d'explorer l'espace des solutions, des chromosomes sont sélectionnés dans la population existante et de nouveaux individus sont créés par des opérations de croisements et de mutations, créant ainsi une nouvelle population (la progéniture). Les croisements permettent de conserver les gènes permettant d'obtenir de bons scores chez les parents, tandis que les mutations permettent d'explorer l'espace des solutions mais également d'éviter une convergence trop rapide en explorant des solutions plus < lointaines >, et maintiennent ainsi une certaine diversité. Les individus les plus forts de la progéniture (i.e. ceux possédant le meilleur score à la fonction objectif) sont ensuite sélectionnés pour former la nouvelle génération. Plusieurs générations sont ainsi créées, jusqu'à obtention d'une population finale. Cette population finale est obtenue quand un nombre d'itérations maximum est atteint, quand un délai est écoulé ou encore lorsqu'un critère de qualité est atteint. Celle-ci sera alors composée d'individus ayant de meilleurs scores à la fonction objectif que les individus initiaux (16). Donc son mise en œuvre nécessite :

Le codage : Le codage est la première étape dans la résolution d'un problème utilisant les algorithmes génétiques. Comme dans la biologie, chaque individu de la population est codé par un chromosome. En algorithme génétique les solutions sont codées par une structure de données contenant l'ensemble des variables de décision liées au problème. Ainsi, le codage est une représentation contenant toute l'information nécessaire à la description d'une solution (chromosome) dans l'espace de recherche . En fonction du problème étudié, les codes utilisés peuvent être binaires ou réels(12)(13) .

Génération de la population initiale : Ce mécanisme doit être capable de produire une population d'individus non homogène qui servira de base pour les générations futures. Le choix

de la population initiale est important car il peut rendre plus ou moins rapide la convergence vers l'optimum global. Dans le cas où l'on ne connaît rien du problème à résoudre, il est essentiel que la population initiale soit assez bien répartie sur tout le domaine de recherche. Une population trop petite évoluera probablement vers un optimum local intéressant alors qu'une population trop grande sera inutile car le temps de convergence sera excessif. La taille de la population doit être choisie de façon à réaliser un bon compromis entre temps de calcul et qualité du résultat (10)(30).

Fonction d'évaluation (Fitness) : La fitness est la pièce maîtresse dans le processus d'optimisation. C'est l'élément qui permet aux algorithmes génétiques de prendre en compte un problème donné. Pour que le processus d'optimisation puisse donner des bons résultats, il faut concevoir une fonction de fitness permettant une évaluation pertinente des solutions d'un problème sous forme chiffrée. Cette fonction est déterminée en fonction du problème à résoudre et du codage choisi pour les chromosomes. Pour chaque chromosome, elle attribue une valeur numérique, qui est supposée proportionnelle à la qualité de l'individu en tant que solution. Le résultat renvoyé par la fonction d'évaluation va permettre de sélectionner ou de refuser un individu selon une stratégie de sélection (30).

Sélection : La sélection a pour objectif d'identifier les individus qui doivent se reproduire. Cet opérateur ne crée pas de nouveaux individus mais identifie les individus sur la base de leur fonction d'adaptation, les individus les mieux adaptés sont sélectionnés alors que les moins bien adaptés sont écartés. La probabilité de sélectionner un individu donné est souvent traduite par le rapport entre la valeur de sa fonction d'adaptation et la somme de toutes les fonctions d'adaptation de la population. Il existe plusieurs techniques de sélection, nous en développons trois : la sélection par roulette biaisée (roulette wheel selection), la sélection par tournoi (tournament selection) et la sélection par rang (ranking selection) (13).

• **Sélection par roulette de loterie (casino) :** Où chaque chromosome sera dupliqué dans une nouvelle population proportionnellement à sa valeur d'adaptation. Son principe est d'associer à chaque individu une portion d'une roue dont la surface dépend uniquement de sa valeur fitness. Chaque individu occupe un secteur de la roulette dont l'angle est proportionnel à sa probabilité de sélection. On actionne N fois la roulette, afin de définir les parents qui assureront la reproduction.

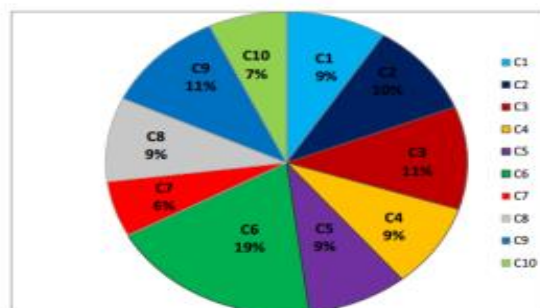


Figure II.7 : Sélection par la méthode de la roulette de casino .(14)

- **Sélection par tournoi** : On tire deux individus aléatoirement dans la population et on reproduit le meilleur des deux dans la nouvelle population. On répète la procédure jusqu'à ce que la nouvelle population soit complète(14) .

- **Sélection par rang** : La sélection par rang de classement commence par trier les chromosomes selon un ordre croissant de la fonction d'évaluation et elle affecte le rang de chacun dans la population (le plus mauvais aura le rang 1). Cette méthode est similaire à celle de la roulette, mais les portions sont en relation avec le rang plutôt qu'avec la fonction d'évaluation. La sélection par rang améliore un peu la chance de sélection des moins bons individus mais sa convergence vers la bonne solution est lente (12).

Croisement : Le croisement est vu comme l'opérateur d'exploration essentiel des algorithmes génétiques appliqué avec la probabilité pc souvent supérieure à 0,5. Son rôle consiste à combiner les génotypes de deux individus pour en obtenir deux nouveaux, en échangeant un ou plusieurs fragments des deux génotypes (13)(40) .

On distingue plusieurs croisements possibles, dont les plus utilisés sont :

- **Le croisement à 1 point** : un seul fragment est échangé selon un point de coupure choisi aléatoirement ;

- **Le croisement à 2 points** : deux fragments sont échangés selon 2 points de coupure choisis aléatoirement ;

- **Le croisement uniforme** : On échange les bits à chaque position indépendamment avec une probabilité de 0.5. Il peut être vu comme un croisement multipoints dont le nombre de coupures est déterminé aléatoirement au cours de l'opération.

La figure suivante représente les opérateurs de croisement :

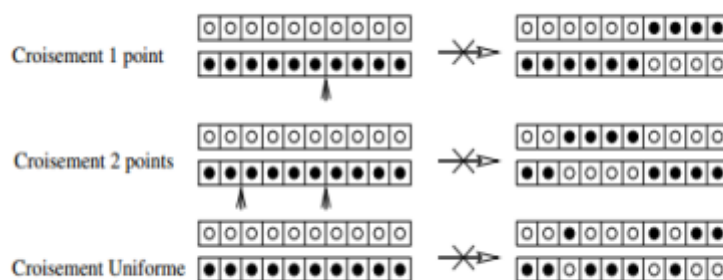


Figure II.8 : Les divers croisements binaire .(40)

Mutation : Le rôle de cet opérateur est de modifier aléatoirement, avec une certaine probabilité, la valeur d'un composant de l'individu. Dans le cas du codage binaire, chaque bit $a_i = \{0, 1\}$ est remplacé selon une probabilité p_m par son inverse $a_i = 1 - a_i$. Nous tirons ainsi pour chaque bit un chiffre aléatoire entre 0 et 1 et si ce chiffre est inférieur à p_m alors la mutation s'opère. Tout comme plusieurs lieux de croisement peuvent être possibles, nous pouvons très bien admettre qu'une même chaîne puisse subir plusieurs mutations. Son rôle est d'apporter la diversité génotypique nécessaire à l'exploration de l'espace de recherche (41)(15).

Remplacement : Le remplacement consiste en l'incorporation des nouvelles solutions dans la population courante. Le remplacement est dit total si l'ensemble des solutions nouvelles sont incorporées, sinon il est dit partiel. Lorsque la nouvelle population n'est constituée que de nouvelles solutions, on parle d'algorithme génétique générationnel. Dans le cas où l'algorithme n'est pas générationnel, des stratégies sont mises en place pour sélectionner les individus à incorporer dans la nouvelle population. En générale ces stratégies sont élitistes est à dire qu'elles privilégient l'incorporation des meilleures solutions dans la population de la nouvelle génération. En notant N le nombre d'individus de la population, on peut distinguer deux types d'élitismes (15) :

- **Élitisme fort** : Les N meilleurs individus parmi l'ensemble des individus enfants et parents sont sélectionnés ;

- **Élitisme faible** : La nouvelle population est obtenue en sélectionnant les K individus parents ayant la meilleur fitness puis les $N - K$ individus enfants ayant la meilleur fitness. K étant un entier fixé.

Critère d'arrêt : Différentes politiques sont possibles pour décider quand arrêter l'algorithme car l'algorithme génétique, comme tous les algorithmes évolutionnaires, sont dans l'absolu sans fin. La politique la plus courante et la plus simple est sans conteste d'effectuer un nombre prédéfini d'itérations mais d'autres sont possibles. On peut distinguer trois grandes familles de critères d'arrêt (30):

- Le temps ou le nombre d'itérations voulu est atteint : il représente la majeure partie des critères d'arrêt employés. En effet, ils sont très faciles à mettre en œuvre. Le temps est lié au nombre d'itérations suivant la taille des données ;

- La fonction d'évaluation est constante depuis quelque temps. Cela permet de caractériser un algorithme qui n'arrive pas à trouver de meilleures solutions et qui est dans une configuration de minimums locaux ;

- La population est dominée par quelques individus. Ce critère est en général assez complexe et peu fiable. Il est très rarement employé.

La Figure représente l'algorithme génétique :

- a. **[Initiation]** Générer aléatoirement une population de n individus (les solutions du problème).
- b. **[Fitness]** Evaluer la fonction d'adaptation $f(x)$ de chaque individu x dans la population.
- c. **[Nouvelle population]** Créer la nouvelle population en répétant les étapes suivantes jusqu'à ce que la population soit complète.
- d. **[Sélection]** Sélectionner deux parents à partir de la population selon la valeur d'adaptation (La meilleure valeur a plus de chance d'être sélectionnée).
- e. **[Crossing-over]** Appliquer l'opérateur de croisement sur les parents avec la probabilité de croisement associée pour donner des enfants. S'il n'y a pas de croisement, les enfants sont la copie identique des parents.
- f. **[Mutation]** Appliquer l'opérateur de mutation sur les enfants avec la probabilité de mutation associée.
- g. **[Accepter]** Placer les enfants dans la nouvelle population.
- h. **[Remplacer]** Utiliser la population générée pour exécuter l'algorithme.
- i. **[Test]** Si la condition d'arrêt est satisfaite, stop, et retourner la meilleure solution.
- j. **[Boucle]** Aller à l'étape b.

Figure II.9 : Pseudo code de l'algorithme génétique .(50)

II.9. Conclusion

L'optimisation est un vaste domaine qui fait intervenir plusieurs types de compétences, c'est pour ça il faut :

- Comprendre le schéma général pour attaquer un problème : Analyse, Modélisation, Résolution, Interprétation, Application ;
- Se ramener si possible à un problème type afin de profiter de l'expérience accumulée ;
- Suivre un fonctionnement itératif et tester abondamment.

Nous avons présenté dans ce chapitre plusieurs méthodes de résolution exactes et approchées. Il est nécessaire de faire appel à des heuristiques permettant de trouver de bonnes solutions approchées. Donc pour résoudre un problème, on doit choisir les méthodes adéquates qui peuvent lui être adaptées.

CHAPITRE III :

Algorithme Firefly (luciole) pour la résolution du problème de covoiturage régulier

III.1. Introduction

Le covoiturage offre plusieurs avantages, ces avantages peuvent être financiers, sociaux environnementaux ou peuvent simplement être la réduction du nombre de voitures sur les routes.

Les avantages financiers sont évidents, si deux personnes partagent une voiture pour un voyage, elles partagent également les coûts du carburant. Pour les voyages fréquents, être à la mesure de réduire les coûts peut être très intéressant, c'est une autre façon de rencontrer des personnes et de rendre le trajet plus agréable. Pour les entreprises, un tel projet crée une dynamique interne qui rayonne rapidement vers l'extérieur.

Après plusieurs années de développement rapide, le covoiturage a déjà été considéré comme un important service de transport alternatif à travers le monde. Dans le but de réduire le trafic et d'encourager la mise en commun de la voiture, certains pays ont introduit des voies de véhicules d'occupation élevée où seuls les véhicules avec deux passagers ou plus sont autorisés à conduire.

Dans ce chapitre nous allons présenter les différentes étapes de fonctionnement des algorithmes FireFly

qui sont la base de notre travail dans le cadre PFE .

III.2. Algorithme des Lucioles (FireFly Algorithm)

III.2.1. 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 (c13).

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 [9] est inspiré par l'atténuation de la lumière sur la distance et l'attraction mutuelle mais il considère toutes les lucioles comme unisexes(c1).

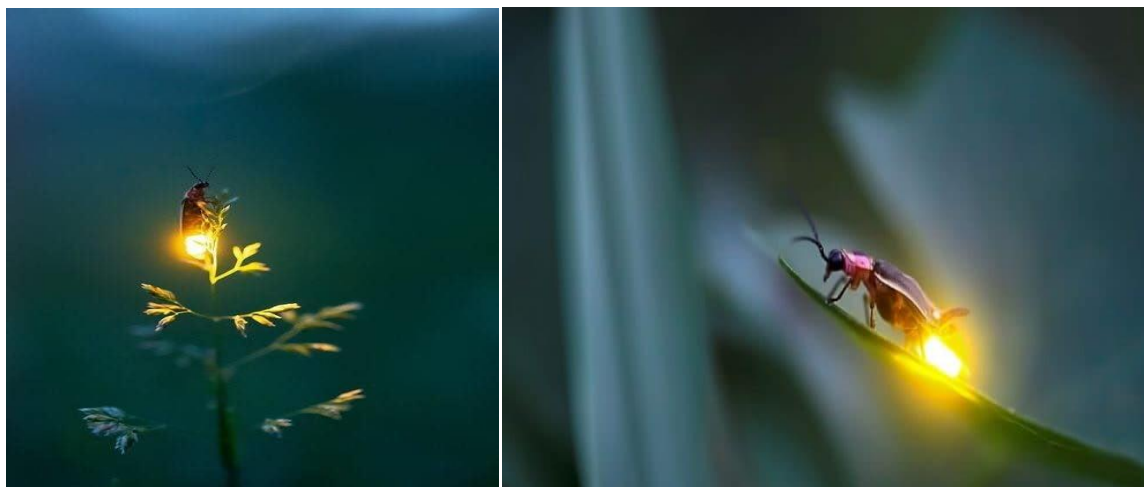


Figure III.1 : Les Lucioles.

III.2.2. Principe de fonctionnement de l'algorithme des Lucioles

L'algorithme Firefly 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 (c14)(c15)(c16):

- 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 métaheuristiques 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'attrance 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 :

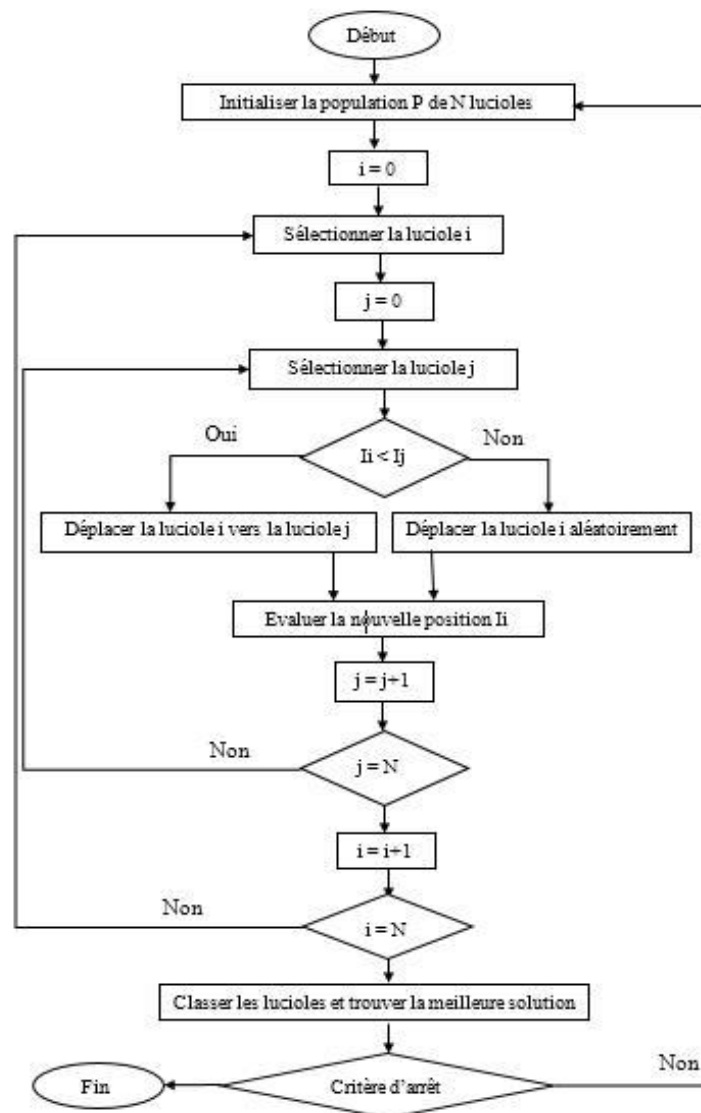


Figure III.2 : Organigramme de fonctionnement d'un algorithme FireFly(c22).

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

```

Générer une population de lucioles  $x_i$  ( $i = 1, 2, \dots, n$ )
Définir l'intensité de lumière  $I$  à un point  $x_i$  par la fonction objective  $f(x_i)$ 
 $t = n$  ;
Tant que ( $t < \text{Max Génération}$ )
  Pour  $i = 1$  jusqu'à  $n$ 
    Pour  $j = 1$  jusqu'à  $n$ 
      Si ( $f(x_i) < f(x_j)$ )
        Déplacer la luciole  $x_i$  vers la luciole  $x_j$ 
      Sinon déplacer  $x_i$  aléatoirement
    Fin Si
    Evaluation des nouvelles solutions et mettre à jour l'intensité de lumière
  Fin Pour j Fin Pour i
  Classer les lucioles et trouver la meilleure solution
   $t++$  ;
Fin Tant que
Visualiser les résultats

```

Figure III.3: L'algorithme FireFly (c19) .

En se basant sur ces trois règles, l'algorithme Firefly se présente comme suit :

1. Génération de la population initiale par un algorithme d'optimisation de regroupement où chaque luciole parente une solution,
2. Chaque luciole est évaluée en fonction de la luminosité, fonction de distance,
3. Comparaison : Si ($I_i < I_j$) La luciole la moins lumineuse sera attirée par la plus lumineuse,
4. Déplacer la luciole i vers la luciole j ,
5. Evaluation des nouvelles solutions et mettre à jour l'intensité de lumière,
6. Classer les lucioles et trouver la meilleure solution(c1).

III.2.3. 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 (c21).

III.2.4. Fonction d'évaluation

La fonction d'évaluation (fitness) en terminologie anglo-saxonne ou de coût, attribue à chaque luciole une valeur numérique qui représente un coût de performance, elle est utilisée pour

coder la luminosité des Fireflies. Grâce 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 :

a) 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.

b) 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.

Critère1

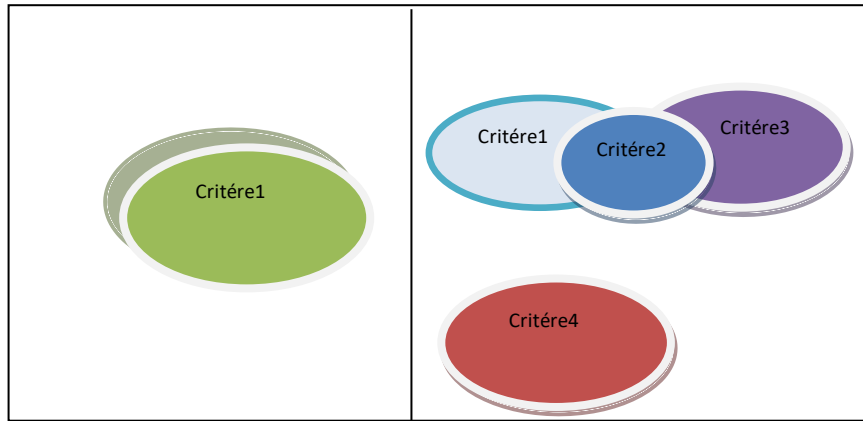


Figure III.4: Fonctions mono critère et multi critère(c1).

III.2.5. 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.

III.2.6. Déplacement et mise à jour

À chaque nouvelle étape itérative, la luminosité et l'attraction 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 (C15).

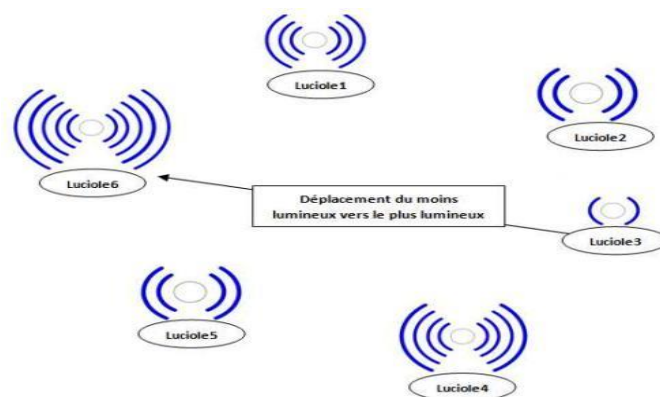


Figure III.5 : Déplacement des lucioles dans une itération(c21).

III.2.7. Critère d'arrêt

Les étapes précédentes (déplacement et mise à jour) appliquées d'une manière itérative, cette boucle s'arrête 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.

III.3. Paramétrages des algorithmes Firefly

La mise en œuvre de l'algorithme firefly ne pose pas de grandes difficultés par contre le paramétrage de l'algorithme, plutôt, le choix des valeurs est l'étape critique. Ces paramètres dépendent étroitement du type de problème à résoudre, le plus souvent les valeurs de ces paramètres sont réglées en fonction des résultats expérimentaux obtenus.

Dans notre algorithme "Firefly Algorithm" on prend en compte les paramètres suivants :

III.3.1. Nombre de luciole

Le nombre de lucioles aussi appelé la taille de la population initiale, à une influence directe sur l'algorithme Firefly pour cela il est très important de bien choisir ce paramètre pour garantir un meilleur compromis entre la qualité de la solution et la rapidité de l'exécution.

D'après' les différents tests effectués dans (c17), on constate que plus la taille de la population est

grande, sa diversité augmente est donc la qualité de solution est meilleure.

Par conséquent si le temps d'exécution de l'algorithme augmente, il affecte l'efficacité de l'algorithme.

Par contre si le nombre de luciole est petit, il y aura alors une probabilité de converger

vers un optimum local et donc il est plus efficace d'avoir un nombre important de lucioles pour assurer une diversité et éviter le problème des minima locaux.

III.3.2. Attraction de luciole

Dans notre l'algorithme, il ya deux valeurs importantes: la variation de l'intensité lumineuse et la formulation de l'attractivité (c18) .

L'attraction de luciole est proportionnelle à l'intensité de la lumière vue par les lucioles adjacentes (c14).

III.3.3. Attraction initiale

C'est l'attractivité quand la distance entre deux lucioles = 0. En général il s'agit d'un paramètre $\beta_0 \in [0,1]$.

Nous considérons deux valeurs limites de β_0 : $\beta_0= 0$ indique une recherche aléatoire non coopérative distribuée.

$\beta_0= 1$ signifie que la recherche coopérative locale où brillante des lucioles détermine les positions des autres lucioles dans son propre quartier [15].

III.3.4. La distance

La distance entre deux lucioles est un paramètre très important, il est évalué de différentes manières. Pour notre algorithme, la distance entre deux lucioles est cartésienne adoptée dans un espace D-dimensionnel.

la distance entre deux lucioles i et j à des positions x_i et x_j est définie par la distance Cartésienne

suivante :

$$r_{ij} = ||x_i - x_j|| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (\text{III.1})$$

Où $x_{i,k}$ représente le $k^{\text{ème}}$ composant spatiale de la coordonné x_i de la luciole i. Dans un repaire 2-D, la formule r_{ij} devient :

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (\text{III.2})$$

III.3.5. Coefficient d'absorption

Le coefficient d'absorption γ contrôle la variation de l'attractivité en fonction de la distance entre deux lucioles communiquées. Il est dans l'intervalle $[0, \infty]$.

$\gamma=0$ correspond à aucun changement, pas de variation ou attractivité constante, $\gamma = \infty$, correspond à une recherche aléatoire complète.

Nous préférons garder la valeur de $\gamma \in [0,1]$, $\gamma = 1$ entraîne une attractivité proche de zéro qui est encore équivalente à la recherche aléatoire complète.

Ce coefficient d'absorption personnalisé pourrait être basé sur la "longueur caractéristique" de l'espace de recherche optimisé(c1).

III.3.6. Nombre de génération

La convergence vers la solution optimale globale n'est pas garantie dans tous les cas même si les expériences dénotent la grande performance de la méthode. De ce fait, il est fortement conseillé de doter l'algorithme d'une portée de sortie en définissant un nombre maximum d'itération(c1).

III.4. Fonction objective

Le terme fonction objective (fitness function en terminologie anglo-saxonne) est utilisé en optimisation mathématique, désigné par une fonction qui sert de critère pour déterminer la meilleure solution à un problème d'optimisation. Concrètement, elle associe une valeur à une instance d'un problème d'optimisation. Le but du problème d'optimisation est alors de minimiser

ou de maximiser cette fonction jusqu'à l'optimum, par différents procédés.

Dans notre cas, l'objectif est de minimiser la quantité de véhicules utilisés et la distance totale parcourue par tous les utilisateurs, ainsi que le cout de déplacement (Essence, péage, ...).

Le covoiturage régulier peut être considéré comme une combinaison d'un problème de clustering et d'un problème de routage.

Ce travail consiste à utiliser la métaheuristique FireFly Algorithm (FFA) pour résoudre le problème de covoiturage régulier de grande taille et ex-tension et l'obtention des résultats plus efficace.

Cependant, nous combinons ces deux objectifs dans quatre fonctions objectives(c1) :

III.4.1. Avant déplacement

Nous savons que l'intensité de la lumière à une distance particulière r de la source de lumière obéit à la loi des carrés inverses. C'est-à-dire que l'intensité de la lumière I diminue à mesure que la distance r augmente en fonction de $I \propto 1/r^2$. Pour cela chaque luciole calcule la luminosité des autres lucioles par rapport à elle avec la formule suivante [Xin09] :

$$I(r) = \frac{1}{r^2} \quad (\text{III.3})$$

r représente la distance entre la luciole i et la luciole j .

Autrement, pour qu'une luciole calcule sa luminosité, elle divise la somme des aluminosité par rapport aux autres lucioles sur le nombre des membres du sous-groupe moins un :

$$I(r) = \frac{p_1/r_2}{K-1} \quad (\text{III.4})$$

k représente le nombre des membres du sous-groupe.

III.4.2. Après déplacement

Comme mentionné ci-dessus l'intensité de la lumière diminue avec la croissancedeladistancepar rapport à la source. Ce qui fait que l'attraction peut varier selon le degré d'absorption.

Pour simplifier, l'intensité de la lumière $I(r)$ varie en fonction de la $I(r) = I_s/r^2$ où I_s est l'intensité à la source. Pour une valeur constante de γ , l'intensité varie en fonction de la distance r , ce qui donne :

$I = I_0 e^{-\gamma r}$, où I_0 est l'intensité de la lumière de la source.

La combine des deux effets de la loi carrée inverse et l'absorption peut être approximer avec la formule Gaussienne suivante [Xin09] :

$$I(r) = I_0 e^{-\gamma r^2} \quad (\text{III.5})$$

La formule (III.5) est utilisée pour qu'une luciole calcule sa luminosité après son déplacement vers une autre luciole plus lumineuse.

r représente la distance parcourue d'une luciole i à une autre j .

D'un autre côté, après le déplacement d'une luciole sa luminosité varie par rapport aux autres lucioles, donc sa luminosité par rapport à ces dernières est calculée comme suit :

$$I(r) = I_j e^{-\gamma r^2} \quad (\text{III.6})$$

Où I_j représente la luminosité de la luciole vers laquelle la luciole i a déplacé par rapport aux autres lucioles. r représente la distance totale parcourue par une luciole i d'une luciole à une autre(c1).

III.5 . Vue artificiel VS Vue réelle

Vue artificiel	Vue réelle
Lucioles	Individus
Luminosité	Fonction objective
Nombre de lucioles	Nombre d'individus

Table III.1 : Vue réelle vs Vue artificielle.

III.6 . Benchmarks

benchmark pour le problème de covoiturage régulier , nous avons utiliser notre propre benchmark qui provient à l'origine du Pickup and delivery Problèmes with Time Windows (PDPTW) par Li et Lim [2003] qui est un problème similaire a notre problème . le benchmark se compose de 9 attribut , le premier attribut indique l'ID de la personne ; le deuxième attribut indique sa position dans l'axe des(X) et le troisième attribut dans l'axe des (Y) , le quatrième n'est pas utilisé pour notre problème le cinquième et le sixième indique respectivement le temps de départ et d'arrivée de chaque individus la première ligne de notre benchmark nous fournit des informations sur la destination.

III.7. Simulation et Résultat

Instance	Size	Meilleur résultat (Temps réponse)	de Distance parcourue	Nombre d'itérations
1	100	28 sec	3698	50
2	80	6 sec	2627	20
3	200	36 sec	7719	20
4	50	10 sec	1696	100

Table III.2 : Résultats expérimentaux les individus.

III.8. Conclusion

L'algorithme des lucioles "Firefly Algorithm" permet de fournir rapidement des solutions qui sont proche de la solution optimale grâce aux mouvements des lucioles en fonction d'attractivité et d'intensité qui est défini par la fonction objectif et qui est le centre de tous les calculs. Les emplacements changent a chaque itération et donc le changement des solutions aussi afin de déterminer le meilleur compromis entre la qualité de la solution et la rapidité de l'algorithme.

Dans ce chapitre, nous avons proposé une approche basée sur les « FireFly » pour minimiser le nombre de voiture sur la route et la distance parcourue globale.

Présentation de l'application

Outils de développement Parmi les différents outils de développement, nous avons choisi ces outils que nous avons utilisés pour réaliser notre travail.

- **Le langage Java**

Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.

La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java(c25).

La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que Unix, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plateformes et Framework associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.

Comme nous avons utilisé la bibliothèque javafx dans notre application voyons alors quelques notions sur ce dernier.

- **JavaFX**

JavaFX est une technologie créée par Sun Microsystems qui appartient désormais à Oracle, à la suite du rachat de Sun Microsystems par Oracle le 20 avril 2009. Avec l'apparition de Java 8 en mars 2014, Javafx devient la bibliothèque de création d'interface graphique officielle du langage Java, pour toutes les sortes d'application (applications mobiles, applications sur poste de travail, applications Web), le développement de son prédécesseur Swing étant abandonné (sauf pour les corrections de bogues). [26] Javafx est désormais une pure API Java (le langage de script spécifique qui a été un temps associé à Javafx est maintenant abandonné).

Javafx contient des outils très divers, notamment pour les médias audio et vidéo, le graphisme 2D et 3D, la programmation Web, la programmation multi-fils etc. Le SDK de Javafx étant désormais intégré au JDK standard Java SE, il n'y a pas besoin de réaliser d'installation spécifique pour Javafx.

- **Historique des versions**

JavaFX est sortie fin 2008, conçu à la base pour concurrencer Silverlight de Microsoft et Flex d'Oracle. Pour JavaFX 1, on devait utiliser le langage de script JavaFXScript. Ce langage a été abandonné dans la seconde version sortie en mai 2011.

JavaFX est compatible pour Windows à partir de XP, pour OS X et Linux. Il existait une version pour appareil mobile. JavaFX Mobile était compatible sur Android, Windows Mobile (pas Windows Phone) et d'autres OS propriétaire. Le support de ce dernier a aussi été abandonné lors de la sortie de JavaFX 2. A partir de cette version, JavaFX est désormais compatible pour iOS et Android.



La dernière version est JavaFX 8 qui est désormais intégré à Java SE 8. De plus, Oracle ne développe plus les fonctionnalités de Swing pour se concentrer sur cette dernière API.

JavaFX peut cohabiter avec Swing et AWT. Ainsi si le développement d'une application a commencé en Swing, les nouvelles interfaces peuvent être ajoutés avec JavaFX (c23) (c24).

- **Scene Builder**

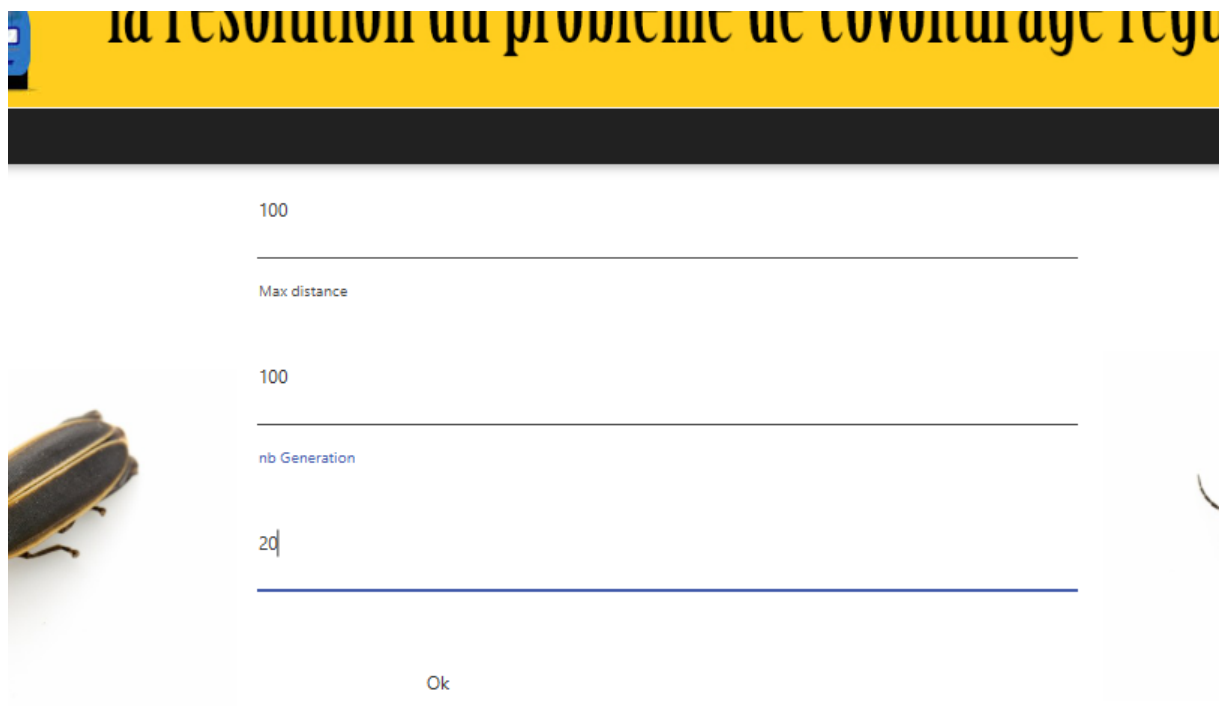
Scene Builder est un outil interactif de conception d'interface graphique pour JavaFX. Créé par Oracle, il permet de construire rapidement des interfaces utilisateurs sans avoir besoin de (savoir) coder.

Le logiciel est décliné en deux versions : l'une (8.x) destiné à JavaFX 8 et l'autre (9.0 et +) pour JavaFX 9 et plus

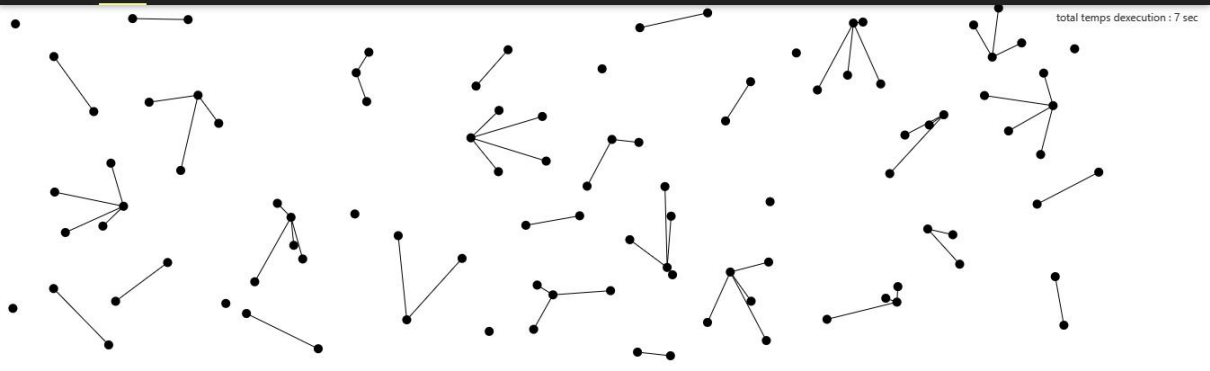
- **Fonctionnement de javafx**

Des projets libres complètent Javafx en fournissant des composants absents de Javafx. Javafx utilise la métaphore d'un théâtre pour modéliser l'application graphique. Une étape (définie par la classe `javafx.stage.Stage`) représente le conteneur de niveau supérieur (fenêtre). Les contrôles (ou composants) individuels sont contenus dans une scène (définie par la classe `javafx.scene.Scene`).

Nous présentons dans ce qui suit quelques captures d'Écrans du fonctionnement du system mis en place :



Interface principale de notre application de 100 individus



Visualisation des groupes de 100 individus

200

Max distance

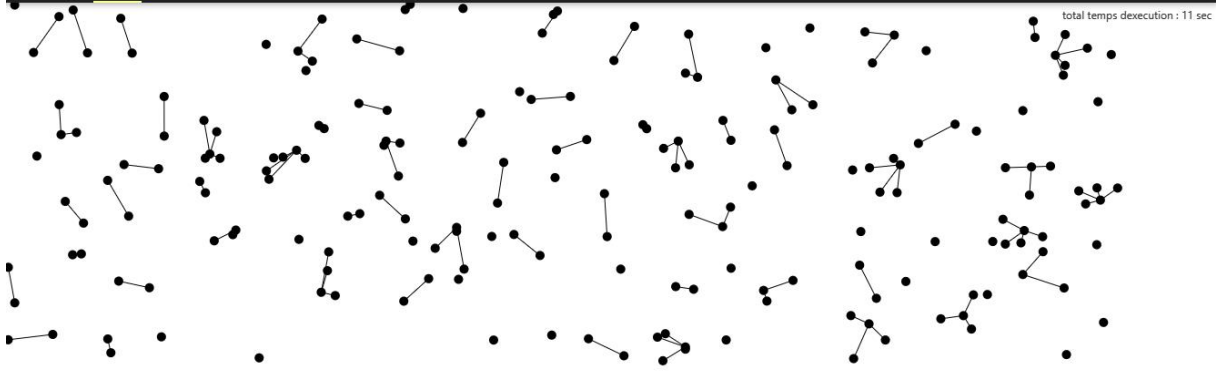
50

nb Generation

20

Ok

Interface principale de notre application de 200 individus



Visualisation des groupes de 200 individus

50

Max distance

100

nb Generation

100

Ok




Interface principale de notre application de 50 individus



Visualisation des groupes de 50 individus

conclusion générale

L'utilisation des moyens de transport collectifs offrant énormément d'avantages à l'utilisateur, les individus ont de plus en plus tendance à aller vers ces moyens les préférant parfois à leur propre voiture. Toutefois, l'automobile occupe une grande place dans leurs vies et reste toujours une source de mobilité très importante et essentielle.

Le covoiturage est devenu ces 10 dernières années un nouveau réflexe de déplacement pour des millions de personnes. C'est devenu une façon de voyager très populaire.

Le covoiturage est une pratique qui permet d'économiser sur les frais de transport. En partageant une même voiture pour un trajet donné, les frais sont divisés entre les passagers. Le covoiturage est un excellent moyen de réduire les embouteillages qui entraînent les émissions de gaz à effet de serre.

Le covoiturage s'annonce alors comme une solution clé alliant les avantages de l'un et l'autre et a ainsi fait l'objet de plus d'un projet. Compte tenu d'une étude élaborée de ce concept, Ce mémoire s'est concentrée sur la mise en place d'une métaheuristique pour résoudre le problème de covoiturage régulier.

Dans Ce mémoire nous avons présenté :

1. Un état de l'art, et quelques travaux antérieurs en rapport avec notre sujet.
2. Les différentes méthodes d'optimisation heuristique & metaheuristique (bio-inspirés) sont présentées.
3. La conception et la modélisation du problème.
4. Une plateforme pour le test, la démonstration, l'évaluation et la comparaison avec des approches pour résoudre le problème du covoiturage régulier.

Comme perspective de notre travail, nous comptons appliquer la méthode proposée sur des cas de covoiturage régulier réel. Plusieurs sites de grandes entreprises s'installent en Algérie, il serait, donc intéressant de voir les résultats de notre approche sur un cas réel.

Bibliographie

(1) Manel Sghaier. RS2.7 : Combinaison des techniques d'optimisation et de l'intelligence artificielle distribuée pour la mise en place d'un système de covoiturage dynamique, PhD thesis, Ecole Centrale de Lille, France, 2011.

(2) **Feki, Mohamed Firas.** *Optimisation distribuée pour la recherche des itinéraires multiopérateurs dans un réseau de transport co-modal.* Lille : Ecole Centrale de Lille, 2010. p.160, Thèse de doctorat. tel-00604509, version 1.

(10). Sarra. Bouallagui. Techniques d'optimisation déterministe et stochastique pour la résolution de problèmes difficiles en cryptologie, institut national des sciences appliquées de rouen. These du doctorat, 2010.

(11). Sarra. Bouallagui. Techniques d'optimisation déterministe et stochastique pour la résolution de problèmes difficiles en cryptologie, institut national des sciences appliquées de rouen. These du doctorat, 2010.

(12). Imen. AyachiHajjem. Techniques avancées d'optimisation pour la résolution du problème de stockage de conteneurs dans un port, école centrale de lille. These du doctorat, 2012.

(13). Hanaa. Hachimi. Hybridations d'algorithmes métaheuristiques en optimisation globale et leurs applications, institut national des sciences École mohammadia d'ingénieurs appliquées de rouen. These du doctorat, 2013.

(14). Zineddine Fridjat and Mohammed Elhadi Tamma. Application des algorithmes génétiques à l'optimisation de la production énergie active dans réseau électrique, université d'el-oued. Mémoire de Master, 2014.

(15). Sophie. Jacquin. Hybridation des métaheuristiques et de la programmation dynamique pour les problèmes d'optimisation mono et multi-objectif : Application à la production d'énergie, laboratoire cristal, école doctorale edspi. These du doctorat, 2015.

(16). Florian Galinier. Des algorithmes génétiques pour générer des modèles diversifiés, université de montpellier. Mémoire de Master, 2016.

(18). **Kammoun, Mohamed Amine.** *Conception d'un système d'information pour l'aide au déplacement multimodal : Une approche multi-agents pour la recherche et la composition des itinéraires en ligne.* Lille : Ecole Centrale de Lille, 2007. p. 181, Thèse de doctorat. tel- 00142340, version 2.

(19). Zgaya, Hayfa. Conception et optimisation distribuée d'un système d'information d'aide à la mobilité urbaine : Une approche multi-agent pour la recherche et la

composition des services liés au transport. Ecole Centrale de Lille. Lille : s.n., 2007. p. 240, Thèse de doctorat. tel-00160802, version 1.

(21). **Kamel Zidi, Slim Hammadi.** *Algorithme génétique avec contrôle des opérateurs pour l'optimisation multicritère d'un déplacement dans un réseau de transport multimodal.* 2005, Revu électronique e-STA, Vol. 2.

(23). **Sidi, M.M. Ould.** *Contribution à l'amélioration des systèmes d'aide à la décision dans le domaine du transport.* Ecole Centrale de Lille. Lille : s.n., 2006. Thèse de doctorat.

(24).BOUARARA Hadj Ahmed : Les méta-heuristiques et les techniques bio-inspirées dans la recherche d'information dans le web social ,Laboratoire de Gestion des Connaissances et des Données Complexes (GeCoDe) Université de Dr Taher Moulay, Saida , 2017.

(30). Abdelhak. Souadkia. Reconnaissance automatique de la parole arabe : Approche évolutionniste, université de guelma. Mémoire de Magistère, 2010.

(36). John H HOLLAND : Genetic algorithms and the optimal allocation of trials In : SIAM Journal on Computing 2.2 (1973), p. 88–105.

(37). Goldberg D E :Genetic algorithms in search, optimization, and machine learning,(1989).

(38). J. C. Setubal and Nalvo F. de Almeida (Eds.) : Advances in Bioinformatics and Computational Biology. Proceedings of the Brazilian Symposium on Bioinformatics, BSB 2013.

(40). Olga. Roudenko. Application des algorithmes évolutionnaires aux problèmes d'optimisation multi objectif avec contraintes, ecole polytechnique x. These du doctorat, 2004.

(41). Thomas Vallée and Murat Yıldızoglu. Présentation des algorithmes génétiques et de leurs applications en économie. Article, 23, 2001.

(50). Eric. Pellerin. Méta-apprentissage des algorithmes génétiques, université du québec à trois rivières. comme exigence partielle de la maîtrise en mathématiques et informatique appliquées, 2005.

(56). Yann Collette, Patrick Siarry. Optimisation multiobjectif. [éd.] Eyrolles. 2212111681.

(57). Othmani, Imed. Optimisation multicritère : fondements et concepts. Université Joseph Fourier de Grenobles. Grenoble, France : s.n., 1998. p. 118, Thèse de doctorat.

(58). Bouyssou, B. Roy and D. Aide multicritère à la décision : méthodes et cas. Economica. 1993.

- (61). **Glover, F.** Future paths for integer programming using surrogate constraints. *Decision Sciences*. 1977, Vol. 8, 1, pp. 156-166.
- (62). Laguna, F. Glover et M. Tabu Serach. Kluwer Academic Publishers. 1997.
- (63). **Fraser, A. S.** Simulation of genetic systems by automatic digital computers. *Australian Journal of Biological Sciences*. 1957, Vol. 10, pp. 484-491.
- (64). **A. Colorni, M. Dorgio et V. Maniezzo.** Distributed Optimization by Ant Colonies. [éd.] édité par F. Valera et al., Elsevier Publishing. *ECAL'91 – First European Conference on Artificial Life*. 1992, pp. 134-142.
- (75). Tao, Chi-Chung. Dynamic Taxi-sharing Service Using Intelligent Transportation System Technologies. International Conference on Wireless Communications, Networking and Mobile Computing. WiCom 2007. 2007, pp. 3209 - 3212 .
- (76). Dridi, Imen Harbaoui. OPTIMISATION HEURISTIQUE POUR LA RÉSOLUTION DU m-PDPTW STATIQUE ET DYNAMIQUE. Lille, France : s.n., 2010. p. 223, Ecole Centrale de Lille.
- (81). W. Jiao, Z. Shi. A dynamic architecture for multi-agent systems. In Proceedings of the 31st International Conference on Technology of Object-Oriented Language and Systems. 1999, pp. 253-260.
- (82). **Kamoun, Mohamed Amine.** *Conception d'un système d'information pour l'aide au déplacement multimodal : Une approche multi-agents pour la recherche et la composition des itinéraires en ligne*. Ecole Centrale de Lille. Lille : s.n., 2007. Thèse de doctorat.
- (83). **Zgaya, Hayfa.** *Conception et optimisation distribuée d'un système d'information d'aide à la mobilité urbaine : Une approche multi-agent pour la recherche et la composition des services liés au transport*. Ecole Centrale de Lille. Lille : s.n., 2007. Thèse de doctorat.
- (84). **Ferber, J.** *Les systèmes Multi-Agents : vers une intelligence collective*. Paris, France : InterEdition, 1995.
- (85). **Bradshaw, J.M.** An introduction to software agents. [éd.] AAAI Press / The MIT Press. *Software agents*. 1997, pp. 3-46.
- (86). **Strugeon, E. Le.** *Une méthodologie d'auto-adaptation d'un système multi-agents cognitifs*. Université de Valenciennes et du Hainaut-Cambrésis. Valenciennes, France : s.n., 1995. Thèse de doctorat.
- (87). **Nwana, H. S.** Software agents : An overview. *Knowledge Engineering Review*. 1996, Vol.11, pp. 1-40.

(88). R.Mandiau, E.G.L.Strugeon, A. Péninou. Organisation et applications des SMA. [éd.]Lavoisier. 2002.

(89). Y. Demazeau, J. Muller. Decentralised artificial intelligence. Elsevier science publisher.1990.

(90). Russell, SJ. et Norvig, P. Artificial intelligence: a modern approach. Englewood Cliffs (New Jersey) : Prentice Hall, 1995.

(91). Florez, R. Towards a standardization of multi-agent system frameworks. ACM Crosswords student magazine. 1999.

(94). **Weiss, G.** *Multiagent systems, a modern approach to distributed artificial intelligence*.s.l. : The MIT Press, 2000.

(103). M. Sghaier, H. Zgaya, S. Hammadi, C. Tahon. DOMARTiC: A Distributed Optimized approach based on the Multi Agent concept to set up a Real Time Carpooling Service. 12^{ème} congrès annuel de la société française de recherche opérationnelle et d'aide à la décision, ROADEF2011. 2011.

(104). F. Sottini, S. Abdel-Naby, P. Giorgini. Andiamo: A Multiagent System to Provide a Mobile-based Rideshare Service, . University of Trento, Ingegneria e Scienza dell'Informazione., Rapport Technique DIT-06-097.

(07) Basile. Martin. Caractérisation du système d'autopartage dans l'agglomération montréalaise et analyse spatio-temporelle de ses différents objets : usagers, stationnements, véhicules , université de montréal. *Mémoire présenté en vue de l'obtention du diplôme de maîtrise des sciences appliquées*, 2007.

(a 3) yuhan guo :Métaheuristiques pour la RÈsolution de Problème de Covoiturage Régulier de Grande Taille et d'une Extension,PhD thesis, Université d'Ar-tois,france,2012.

(a2) http://dante.univ-tlse2.fr/131/1/Faessel_Julien_rapport.pdf

(c1) Gestion de la qualité de service dans un réseau

radio cognitive en utilisant l'algorithme FireFly

(c19)LatifaDekhiciandKhaledBelkadi.Métaheuristiquesdeslucioles discrète pourlesflowshophybrides,universitéu.s.t.o.m.b,oran.

[21] Hadjer. Gahgah. Problème d'emploi de temps : proposition un algorithme bio-inspiré, universite mohamed boudiaf - m'sila. Mémoire de Master, 2018.

(c9) Yang X.S,"Nature-Inspired Metaheuristic Algorithms". Luniver Press, UK. 2008.

(c13) H.K.F. Stanger, J.E. Lloyd., D.M. Hillis, "Phylogeny of North American fireflies

(Coleoptera: Lampyridae)", Vol. 45, Issue 1, 2007, pp. 33–49.

- (c14) Ali Saoucha, N. K. Ghanem, and B. Benmammar. "On applying firefly algorithm for cognitive radio networks." Communications and Vehicular Technology in the Benelux (SCVT), 2014 IEEE 21st Symposium on. IEEE, 2014.
- (c15) Romana CAPOR-HROSIK, Adis ALIHODZIC, Milan TUBA, Mirjana VUKOVIC, Milenko PIKULA, "Firefly Algorithm for Constrained Optimization Problems". ISBN: 978-960-474-330-8.
- (c16) Prabhneet kaur, Taranjot kaur, " A Comparative Study of Various Metaheuristic Algorithms", Guru Teg Bahadur Institute of Technology, GGSIPU, New Delhi.
- (c17) S. E. Merzouk, "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.
- (c18) Xin-She Yang, "Nature-Inspired Metaheuristic Algorithms", 2nd Edition Copyright © 2010 Luniver Press.
- (c22) Mohamed Amine. Beghoura. Segmentation multi sources des images satellitaires par l'algorithme firefly, universite des sciences et de la technologie d'oran mohamed boudiaf. Mémoire de Magistère, 2012.
- (c23) <http://docs.oracle.com/javase/8/javafx/get-started-tutorial/index.html>
- (c24) <http://mikarber.developpez.com/tutoriels/java/introduction-javafx/>
- (c25) "Qu'est-ce que c'est que Java et pourquoi j'en ai besoin", https://www.java.com/fr/download/faq/whatis_java.xml, dernière visite : 21/05/2018.
- (c26) Herbert Schildt, "introducing javafx8 programming", oracle press, new York 2015.