

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mira Abderrahmane de Béjaïa
Faculté des Sciences Exactes
Département de Recherche Opérationnelle



Mémoire de fin de cycle

Pour l'Obtention du Diplôme de Master en Recherche Opérationnelle
Option : Modélisation Mathématiques et Techniques de Décision

*Résolution du problème de tournées de
véhicules par une méta-heuristique :
Application au transport universitaire à Béjaïa*

Présenté Par :

-*M^{elle}* Ahlam OUALI
-*M^{elle}* Lilia ROUBIOU

Soutenu publiquement à l'Université de Béjaïa, le 04/07/2019, devant le jury
composé de :

M ^{me} N. HALIMI-YOUSFI	Présidente	à l'Université de Béjaïa.
M ^r M.S. RADJEF	Encadreur	à l'Université de Béjaïa.
M ^{elle} L. IDRES	Co-Encadreur	à l'Université de Béjaïa.
M ^r S. TOUATI	Examineur	à l'Université de Béjaïa.
M ^{elle} Z. AOUDIA	Examinatrice	à l'Université de Béjaïa.

Année Universitaire 2018 – 2019



Louange A Dieu, le miséricordieux, sans Lui rien de tout cela n'aurait pu être.

Je tiens tout d'abord à remercier le Professeur M^r M.S. RADJEF et M^{elle} L. IDRES pour l'honneur qu'ils m'ont fait en acceptant de nous encadrer. leurs conseils précieux ont permis une bonne orientation dans la réalisation de ce modeste travail.

Je teins également à remercier M^{me} N. HALIMI-YOUSFI d'avoir accepté de présider le jury de ce mémoire.

Je remercie M^{elle} Z. AOUDIA et M^r S. TOUATI d'avoir accepter de faire partie du jury et consacrer leurs temps à la lecture et à la correction de ce mémoire.

Mes remerciements les plus vifs vont tout particulièrement à nos parents.

Enfin, merci à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.



Je dédie ce modeste travail en principe aux personnes les plus proches de ma vie ; mon père et ma mère qui ont été présents pour moi pendant tout mon cursus et ma vie, mais surtout qui ont toujours su trouver les mots qui m'encourageaient et qui me poussaient à aller de l'avant.

Pour tous les instants passés, les fous-rires et le désespoir, la colère et la joie qui ont accompagné ce travail, et même pour sa simple présence, je dédie ce mémoire à mon binôme LILIA.

Je dédie aussi ce travail à toutes ma famille qui était toujours derrière moi pour me fortifier pendant mes moments difficiles, en particulier :

★ Mes soeurs : ASSIA, AMIRA.

★ Mes frères : AHMED, AMINE.

Je veux aussi dédier ce travail à mes chers amis qui ne nous ont jamais oubliés et qui étaient présents en cas de besoin : NASSIMA, WISSAM, KARIM.

Je dédie aussi ce travail, en particulier :

★ Mon mari : NOUREDDINE.

Enfin à toutes personnes qui me sont chères et que j'aime.

AHLAM



Je dédie ce modeste travail en principe aux personnes les plus importantes de ma vie ; ma mère qui a été présente pour moi pendant tout mon cursus et ma vie, mais surtout qui a toujours su trouver les mots qui m'encourageaient et qui me poussaient à aller de l'avant.

*Pour tous les instants passés, les fous-rires et le désespoir, la colère et la joie qui ont accompagné ce travail, et même pour sa simple présence, je dédie ce mémoire à mon binôme **AHLAM** sans laquelle il n'aurait pas pu être réalisé.*

Je dédie aussi ce travail à toutes ma famille qui était toujours derrière moi pour me fortifier pendant mes moments difficiles, en particulier :
★ Mes ancres, mes grands parents.

*Je veux aussi dédier ce travail à mes chers amis qui ne nous ont jamais oubliés et qui étaient présents en cas de besoin : **KARIM, NOUARA, LATIFA, SABRINA.***

Enfin à toutes personnes qui me sont chères et que j'aime.

LILIA

Table des matières

Liste des figures	vi
Liste des algorithmes	vii
Liste des tables	viii
Introduction générale	1
1 Problème de tournées de véhicules	3
Introduction	3
1.1 Problème du voyageur de commerce	4
1.1.1 Formulation mathématique	4
1.2 Définition du VRP	5
1.2.1 Formulation du problème	6
1.3 Variantes du VRP	8
1.3.1 VRP à Contraintes liées à la Flotte de véhicules(CVRP)	10
1.3.2 VRP à Contraintes liées à la Demande des Clients	11
1.3.3 VRP à Contraintes liées aux Dépôts	11
1.3.4 VRP à Contraintes liées aux Temps	11
1.4 Méthodes de résolution	12
1.4.1 Méthodes exactes :	12
1.4.2 Méthodes approchées :	12
Conclusion	14
2 Problème de tournées des autobus scolaires (SBRP)	15
Introduction	15
2.1 Revue de la littérature	16
2.2 Modélisation du problème	19
2.2.1 Modèle	20
Conclusion	22
3 Approches de Résolution	23
Introduction	23
3.1 Approches exactes	24
3.1.1 Programmation Linéaire	24
3.1.2 Programmation dynamique	25

3.1.3	Approche de Branch and X	26
3.2	Approches heuristiques	28
3.2.1	Heuristiques de construction	28
3.2.2	Méthodes à deux phases	30
3.2.3	Heuristiques d'amélioration	30
3.3	Méta-heuristiques	32
3.3.1	Algorithmes génétiques	32
3.3.2	Réduit simulé	33
3.3.3	Méthode de recherche tabou	34
3.3.4	Optimisation par Colonies de fourmis	35
	Conclusion	40
4	Application	41
	Introduction	41
4.1	Présentation de la direction des œuvres universitaires	42
4.2	Position du problème	42
4.3	Collecte de données	43
4.4	Implémentation et résultats	47
4.4.1	Cas fluide	48
4.4.2	Cas congestionné	48
4.4.3	Interprétation des résultats	57
4.5	Comparaison des résultats avec l'organisation actuelle du transport universitaire	57
	Conclusion	58
	Conclusion générale	59
	Bibliographie.	61

Table des figures

1.1	Problème du voyageur de commerce	4
1.2	Exemple de tournées avec 4 véhicules et 20 clients	8
3.1	Illustration du principe de l'algorithme de gain	29
3.2	Avant l'application de l'heuristique 2-opt	31
3.3	Après l'application de l'heuristique 2-opt	31
3.4	Avant l'application de l'heuristique Or-opt	31
3.5	Après l'application de l'heuristique Or-opt	31
3.6	Organigramme du fonctionnement de l'Algorithme ACO	38
4.1	Présentation géographique des arrêts des bus universitaires	46
4.2	Présentation géographique des arrêts des bus universitaires : pris dans notre étude	46

Liste des Algorithmes

1	Algorithme génétique	33
2	Recuit simulé	34
3	Algorithme de recherche tabou	35
4	Algorithme de colonies de fourmis	39

Liste des tableaux

1.1	Paramètres de classification des problèmes VRP	9
4.1	Arrêts universitaires	44
4.2	Arrêts intermédiaires	45
4.3	Arrêts du réseau	47
4.4	Tournées optimales dans le cas fluide	48
4.5	Tournées optimales dans le cas 1 : $T_{max}^k = 1h, n/k = 0.9$	49
4.6	Tournées optimales dans le cas 2 : $T_{max}^k = 1h, n/k = 0.95$	50
4.7	Tournées optimales dans le cas 3 : $T_{max}^k = 1h, n/k = 1$	51
4.8	Tournées optimales dans le cas 4 : $T_{max}^k = 0.5h, n/k = 0.9$	52
4.9	Tournées optimales dans le cas 5 : $T_{max}^k = 0.5h, n/k = 1$	53
4.10	Tournées optimales dans le cas 6 : $T_{max}^k = 0.5h, n/k = 1$	54
4.11	Tournées optimales dans le cas 7 : $T_{max}^k = 1h, n/K = 0.9$	55
4.12	Tournées optimales dans le cas 8 : $T_{max}^k = 1h, n/K = 0.95$	56
4.13	Tournées optimales dans le cas 9 : $T_{max}^k = 1h, n/K = 1$	57
4.14	Tournées actuelles	58

Introduction générale

De nos jours, le problème de transport occupe une place très importante dans la vie des sociétés modernes, raison pour laquelle, divers modèles d'optimisation combinatoire conçus pour l'étude de ces problèmes ont vu le jour.

Le problème de base le plus étudié, est le Problème du Voyageur de Commerce (PVC) dit aussi Traveling Salesman Problem (TSP), qui a pour objectif, de déterminer une tournée acheminée par un véhicule. Tout en respectant un certain nombre de critères tels que : la minimisation du temps ou de la distance totale parcourue. Dans le cas où, on a un ensemble de véhicules prêt à servir un ensemble de demandes et qui doivent faire plusieurs tournées tout en retournant au même point d'origine appelé le dépôt centrale, on parle de problème de tournées de véhicules définit il y a plus de 50 ans. Ce problème est très étudié à cause de l'importance croissante du transport de passagers et de marchandises, il consiste à trouver l'ensemble optimal d'itinéraires pour une flotte de véhicules afin de servir un ensemble donné de clients. Il est qualifié de difficile et appartient à la classe des problèmes NP-DURS.

L'organisation du transport scolaire est courante dans des pays où de petites communes procèdent à une centralisation des infrastructures scolaires. et le service du transport universitaire a pour but d'acheminer des étudiants de leurs résidences à leurs campus et vice versa, selon le nombre d'étudiants, le nombre de campus et le nombre de bus appropriés. La planification d'un tel service s'avère longue et fastidieuse si elle est effectuée manuellement. À première vue, cette problématique s'apparente au "problème de tournées de véhicules". L'objectif de ce travail est la résolution du problème de tournées des bus universitaires appliqué à la ville de Béjaia avec la métaheuristique colonie de fourmis. Ce mémoire est constitué de quatre chapitres et se présente comme suit :

- Le premier chapitre abordera les notions de base concernant les problèmes de tournées véhicules (VRP) et ses différentes variantes.
- Le deuxième chapitre présente une synthèse bibliographique des travaux effectués sur le problème SBRP (School Bus Routing Problem) et la modélisation du problème posé.
- Le troisième chapitre fait l'objet d'une description des méthodes de résolution exactes et approchées pour le problème VRP (vehicule routing problem) et plus exactement l'algorithme de colonie de fourmis appliqué au SBRP.

- Le quatrième chapitre est consacré aux résultats du cas pratique considéré dans ce mémoire.

1

Problème de tournées de véhicules

Contents

Introduction	3
1.1 Problème du voyageur de commerce	4
1.2 Définition du VRP	5
1.3 Variantes du VRP	8
1.4 Méthodes de résolution	12
Conclusion	14

Introduction

Le problème de tournées de véhicules, plus connu sous son nom anglais Vehicle Routing Problem (VRP) est une extension du problème du voyageur du commerce. En fait, la première formulation du problème est attribuable à Dantzig et al, 1954 [14]. Ce problème est très fréquent dans la pratique : les ramassages scolaires, transport en commun, livraison de marchandises à des clients... Dans ce chapitre nous allons introduire le problème de tournées de véhicules VRP (Véhicule Routing Problem), nous rappellerons par la suite les différentes méthodes de résolution, voir les méthodes exactes, les heuristiques et les métaheuristiques.

1.1 Problème du voyageur de commerce

Le problème du voyageur de commerce, ou Traveling Salesman Problem (TSP) naît d'une problématique vécue par des vendeurs ou commerciaux qui se déplacent pour livrer ou rencontrer leurs clients. En effet, un vendeur a pour mission de visiter n villes et retourne à son point de départ après les avoir visitées une et une seule fois chacune. L'objectif du vendeur, qui visite toutes les villes, est de définir un parcours qui minimise la distance totale. Plus formellement, un TSP est modélisé sous la forme d'un graphe où les villes sont représentées par des noeuds, et les liaisons entre elles par des arêtes. Dans le cadre de ce problème, un vendeur part d'un dépôt pour livrer les demandes des clients et rentre après avoir livré le dernier client.

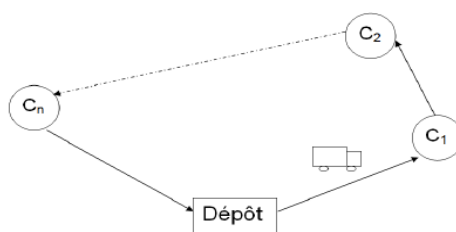


FIGURE 1.1 – Problème du voyageur de commerce

Le problème du TSP a été évoqué pour la première fois par Euler (1736) [1].

1.1.1 Formulation mathématique

Le TSP est défini sur un graphe $G = (V, E)$, où V désigne l'ensemble des sommets, comprenant un ensemble de clients $V' = \{1, \dots, n\}$ et un noeud-dépôt 0. E correspond à l'ensemble des arcs (i, j) avec $i, j \in V$. La formulation mathématique du TSP requiert la définition de C_{ij} le coût du parcours de l'arc (i, j) et d'une variable de décision x_{ij} .

$$x_{ij} = \begin{cases} 1 & \text{si le voyageur traverse l'arc}(i, j); \\ 0 & \text{sinon.} \end{cases} \quad (i, j) \in E; \quad (1.1)$$

Une formulation classique du TSP en programme linéaire en nombres entiers peut être définie comme suit :

Fonction-objectif :

$$\min \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij}. \quad (1.2)$$

Sous contraintes :

$$\sum_{i=1}^n x_{ij} = 1; \quad \forall j = 0, \dots, n; \quad (1.3)$$

$$\sum_{j=1}^n x_{ij} = 1; \quad \forall i = 0, \dots, n; \quad (1.4)$$

$$\sum_{i \in S} \sum_{\substack{j \in S \\ j \neq i}} x_{ij} \leq |S| - 1; \quad \forall S \subset V', |S| \geq 2; \quad (1.5)$$

$$x_{ij} \in \{0, 1\}; \quad \forall i = \overline{1, n}; \forall j = \overline{1, n}. \quad (1.6)$$

La fonction-objectif (1.2) cherche à minimiser le coût total de la tournée parcourue. Les contraintes (1.3) et (1.4) imposent que le voyageur passe une et une seule fois par chaque nœud. L'élimination des sous-tours, c'est-à-dire des trajets passant par un sous-ensemble de nœuds sans retour au dépôt, est assurée par les contraintes (1.5).

1.2 Définition du VRP

Le problème classique de tournées de véhicules consiste à concevoir un ensemble de tournées, pour un nombre fini de véhicules, commençant et finissant à un dépôt et passant par un ensemble de points (souvent nommés clients ou villes). Dans ces tournées, un client doit être desservi une seule fois par un seul véhicule et la capacité de transport d'un véhicule pour une tournée ne doit pas être dépassée.

Les objectifs les plus communs du problème VRP sont la minimisation de la distance totale parcourue par les véhicules ou la minimisation du temps total de parcours des tournées. Toute fois d'autres objectifs peuvent être considérés, on cite[21] :

- la minimisation du nombre de véhicules utilisés ;
- la minimisation du coût total des tournées (en prenant en compte les coûts des véhicules, des chauffeurs,...) ;
- la maximisation des gains engendrés par les tournées ;
- l'équilibrage de charge entre les véhicules,...etc.

Selon Bräysy, en 2001 [5], les objectifs de minimisation du nombre de véhicules et de la distance (ou durée) totale des tournées sont conflictuels : la diminution du nombre de véhicules engendre le plus souvent une augmentation de la distance totale parcourue. Raison pour laquelle on peut traiter le problème de VRP comme un problème multiobjectif.

1.2.1 Formulation du problème

Dans cette section, nous formulerons le problème classique de VRP. Pour cela, considérons un graphe $G = (V, E)$ dans lequel les arcs représentent les axes routiers et les sommets les points de passage de véhicules[3].

La formulation mathématique du problème de tournées de véhicules est donc donnée comme suit :

$V = (v_0, v_1, v_2, \dots, v_{n-1})$ est l'ensemble des n sommets du graphe représentant l'ensemble des clients, avec v_0 représentant le dépôt. On pose $V' = V \setminus \{v_0\}$.

$E = \{(v_i, v_j) | v_i, v_j \in V; i \neq j\}$ est l'ensemble des arcs orientés représentant le trajet entre deux sommets v_i, v_j .

Soit les notations suivantes :

- n nombre de clients (ou sommets) ;
- m nombre de véhicules $m \leq n$;
- d_{ij} distance entre le sommet v_i et le sommet v_j , s'il n'existe pas de chemin entre les sommets v_i et v_j (impasse, rue piétonne, ...), alors on posera $d_{ij} = \infty$;
- t_i^k le temps d'arrêt du véhicule k au sommet v_i ;
- t_{ij}^k le temps mis par le véhicule k pour aller du sommet v_i au sommet v_j ;
- T^k le temps maximal de la tournée du véhicule k ;
- Q^k la capacité du véhicule k (en termes de biens à transporter : nombre de places s'il s'agit de personnes, tonnages s'il s'agit de pondéreux, etc.) ;
- q_i la demande du client i ;
- c_{ij} le coût de l'arête entre les sommets v_i et v_j (coût monétaire ou temps de parcours) ;
- $i = 0, \dots, n$ est l'indice des sommets prédécesseurs ;
- $j = 0, \dots, n$ est l'indice des sommets successeurs, telles que 0 indique le dépôt ;
- $k = 1, \dots, m$ est l'indice des véhicules ;
- Les variables de décision du problème sont les x_{ij}^k , tel que :

$$x_{ij}^k = \begin{cases} 1 & \text{si } (v_i, v_j) \text{ est parcouru par le véhicule } k; \\ 0 & \text{sinon.} \end{cases} \quad (1.7)$$

La fonction à minimiser est donnée comme suit :

$$\min \sum_{i=0}^m \sum_{j=0}^n c_{ij} \sum_{k=1}^n x_{ij}^k. \quad (1.8)$$

Sous contraintes :

$$\sum_{k=1}^m \sum_{i=0}^n x_{ij}^k = 1; \quad \forall j \in V'; \quad (1.9)$$

$$\sum_{k=1}^m \sum_{j=0}^n x_{ij}^k = 1; \quad \forall i \in V'; \quad (1.10)$$

$$\sum_{i=0}^n \sum_{l=0}^n x_{il}^k = \sum_{l=0}^n \sum_{j=1}^n x_{lj}^k; \quad \forall i = 1, \dots, n; \quad (1.11)$$

$$\sum_{i \in S} \sum_{\substack{j \in S \\ j \neq i}} x_{ij}^k \leq |S| - 1; \quad \forall S \subset V', |S| \geq 2, \forall 1, \dots, m; \quad (1.12)$$

$$x_{ij}^k \in \{0, 1\}; \quad \forall 0 \leq i \leq n; 0 \leq j \leq n; 1 \leq k \leq m. \quad (1.13)$$

La fonction-objectif (1.8) cherche à minimiser la somme des coûts de toutes les tournées. Les contraintes (1.9) et (1.10) imposent que chaque client soit desservi une et une seule fois. La contrainte (1.11) assure la conservation du flot, le sommet visité doit impérativement être quitté. La contrainte (1.12) assure l'empêchement des sous-tournées, le véhicule ne doit pas revenir au dépôt avant de terminer sa tournée. La contrainte (1.13) est la contrainte de binarité des variables de décisions $x_{ij}^k[1]$.

Remarque : Selon le problème de VRP considéré, d'autres contraintes pourraient être rajoutées.

Exemple : La Figure 1.2 montre des tournées avec quatre véhicules qui assurent le passage par tous les sommets, en partant et revenant au dépôt.

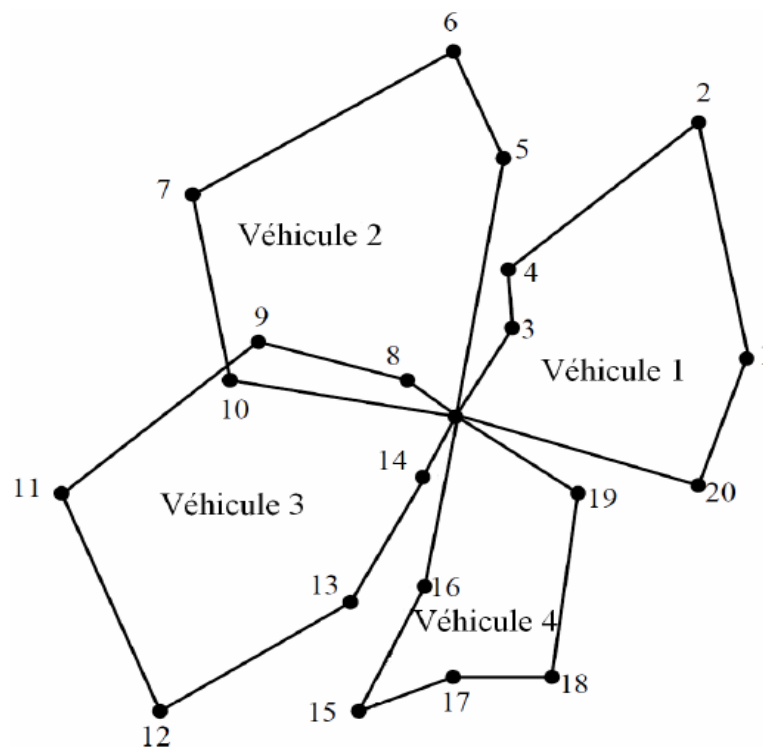


FIGURE 1.2 – Exemple de tournées avec 4 véhicules et 20 clients

1.3 Variantes du VRP

Plusieurs variantes du VRP ont été définies dans la littérature. En effet, le fait d'ajouter, de changer ou de supprimer des contraintes peut nous faire passer d'un problème VRP à un autre[1].

Le tableau suivant résume les paramètres selon lesquels sont classés les problèmes VRP :

Paramètres	Options possibles
Distances entre noeuds	- Euclidiennes ou non - Constantes - Symétriques - Différentes (asymétriques)
Emplacement des véhicules	- Un seul dépôt - Plusieurs dépôts
Types de véhicules	- Homogène - Hétérogène
Taille de la flotte	- Un seul véhicule - Plusieurs véhicules
Nature des demandes	- Déterministe - Stochastique
Longueur maximale d'une tournée	- Imposée et identique pour toutes les tournées - Imposée et non identique pour toutes les tournées - Non déterminée
Temps maximal d'une tournée	- Imposée et identique pour toutes les tournées - Imposée et non identique pour toutes les tournées - Non déterminée
Nombre de livraisons/visites à un client	- Unique - Multiple mais unique pour un type de véhicule
Horaires de livraisons	- Non définis - Définis de manière précise (rendez-vous) - Définis d'un intervalle horaire à un noeud
Nombre de tournées admises par véhicule	- Unique - Multiple
Calcul des coûts de transport	- Fonction de la distance - Fonction du temps - Fonction du nombre de véhicules
Opérations de collectes	- Collectes et livraisons simultanées chez les clients - Soit collecte soit livraison chez un client
Type de client	- Dépendant - Indépendant
Demande des clients	- Statique - Dynamique

TABLE 1.1 – Paramètres de classification des problèmes VRP

Dans ce qui suit on introduit les principales variantes du problème VRP[15].

1.3.1 VRP à Contraintes liées à la Flotte de véhicules(CVRP)

✓ **CVRP** (Capacitated Vehicle Routing Problem)

Un problème CVRP est un problème de tournées de véhicules avec des contraintes de capacités. Les véhicules ont une capacité d'emport limitée (quantité, volume, poids,...). Dans ce cas, une tournée réalisable est une tournée où la somme des demandes de ses clients ne dépasse pas la capacité du véhicule qui l'effectue, ce qui est exprimé par les contraintes suivantes :

$$\sum_{i=1}^n q_i \left(\sum_{j=1}^n x_{ij}^k \right) \leq Q^k; \quad \forall k = 1, \dots, m. \quad (1.14)$$

✓ **VRP-FL** (Vehicle Routing Problem with Full Truckload)

Dans ce problème, le véhicule occupe toute sa capacité, ce qui est exprimé par les contraintes suivantes :

$$\sum_{i=1}^n q_i \left(\sum_{j=1}^n x_{ij}^k \right) = Q^k; \quad \forall k = 1, \dots, m. \quad (1.15)$$

✓ **VRP-HF** (Vehicle Routing Problem with Heterogeneous Fleet)

La flotte des véhicules est hétérogène. Les clients sont affectés à différents type de véhicules de caractéristiques différentes en terme de charge ou temps de route maximum, coûts fixes, et coûts variables en fonction de la distance [23].

✓ **VRPB** (Vehicle Routing Problem with Back)

Chaque tournée commence et se termine au dépôt, ce qui est exprimé par les contraintes suivantes :

$$\sum_{j=0}^n x_{0j}^k = 1; \quad \forall k = 1, \dots, m; \quad (1.16)$$

$$\sum_{i=0}^n x_{i0}^k = 1; \quad \forall k = 1, \dots, m. \quad (1.17)$$

✓ **OVRP** (Open Vehicle Routing Problem)

La différence entre un OVRP et un VRP est que dans le premier les véhicules ne sont pas tenus de retourner au dépôt(respectivement de démarrer du dépôt)[9]. Ce qui est exprimé par les contraintes suivantes :

$$\sum_{j=0}^n x_{0j}^k \neq 1; \quad \forall k = 1, \dots, m; \quad (1.18)$$

$$\sum_{i=0}^n x_{i0}^k \neq 1; \quad \forall k = 1, \dots, m. \quad (1.19)$$

1.3.2 VRP à Contraintes liées à la Demande des Clients

✓ **DVRP** (VRP à Demande Déterministe)

Le DVRP est un problème où la livraison se fait suite à une commande. En effet, le livreur connaît avant son départ du dépôt la quantité à livrer à chacun de ses clients.

✓ **SVRP** (VRP à Demande stochastique)

Un problème VRP est dit stochastique si la demande des clients est aléatoire et incertaine, c'est au moment de livraison que le livreur la découvre. La demande de chaque client est estimée approximativement par une fonction stochastique. La plupart des recherches supposent que les demandes aléatoires suivent une loi de distribution "normal".

✓ **SDVRP** (Split Delivery Vehicle Routing Problem)

Dans ce genre de problèmes, la demande du client peut être satisfaite sur plusieurs tournées. Cette variable est utilisée dans le cas où la demande d'un ou plusieurs clients peut excéder la capacité du véhicule effectuant une tournée [2].

✓ **VRPPD** (Vehicle Routing Problem Pick-up and Deliveries)

Le problème de collecte et de livraison a les mêmes propriétés que le VRP. Dans le VRPPD, on prend en plus en considération le fait que chaque client introduit deux positions géographiquement différentes : la première pour le ramassage du produit et la deuxième pour la livraison de ce dernier.

1.3.3 VRP à Contraintes liées aux Dépôts

✓ **MDVRP** (Multi-Depot Vehicle Routing Problem)

Plusieurs dépôts géographiquement distribués existent. Ce qui fait que les véhicules peuvent s'approvisionner de plusieurs dépôts. Les véhicules partent de n'importe quel dépôt mais reviennent à leurs dépôts de départ.

✓ **1DVRP** (Vehicle Routing Problem)

Un seul dépôt existe, les véhicules doivent s'approvisionner d'un seul dépôt.

1.3.4 VRP à Contraintes liées aux Temps

✓ **VRPTW** (Vehicle Routing Problem with Time Windows)

Le VRPTW est l'un des problèmes les plus étudiés. Dans un VRPTW, de nouvelles contraintes temporelles sont ajoutées : chaque client doit être servi dans un intervalle de temps durant lequel il est disponible pour être visité tout en respectant la durée totale d'une tournée (cette durée est la somme des temps des voyages entre les sommets et les temps d'arrêts à ces derniers) qui est exprimée dans les contraintes

suivantes [15] :

$$\sum_{i=0}^n t_i^k \sum_{j=0}^n x_{ij}^k + \sum_{i=0}^n \sum_{j=0}^n t_{ij}^k x_{ij}^k \leq T_k; \quad \forall k = 1, \dots, m. \quad (1.20)$$

Et toute violation de ces contraintes peut engendrer une pénalité [6]. IL peut aussi être dynamique DVRPTW, dans ce cas il assume la composante dynamique d'arrivées d'événements urgents (nouvelle demande, nouveau client, panne d'un véhicule...)[10].

✓ **PVRP** (Periodic Vehicle Routing Problem)

Le problème de tournée de véhicules périodiques consiste à livrer pour un ensemble de clients, la quantité demandée périodiquement selon une planification prédéfinie.

1.4 Méthodes de résolution

Dans le cadre d'une démarche de résolution des problèmes de VRP, plusieurs méthodes ont été développées dans la littérature. Ces méthodes ont été classées en deux grandes familles : les méthodes exactes, les méthodes approchées [21].

1.4.1 Méthodes exactes :

Le principe des méthodes exactes, également appelé méthodes complètes ou optimales, consiste à énumérer implicitement et le plus efficacement possible toutes les solutions réalisables du problème pour extraire une solution optimale, pour des problèmes de taille relativement petite.

Plusieurs méthodes exactes ont été proposées dans la littérature. Elles sont basées sur les approches de programmation linéaire, de programmation dynamique.

1.4.2 Méthodes approchées :

Ces méthodes sont utilisées pour la résolution des problèmes d'optimisation combinatoire pour lesquels on ne trouve pas de solution optimale en un temps raisonnable. On distingue différentes classes de méthodes approchées.

✓ **Les heuristiques** : Les heuristiques appartiennent à la classe des méthodes approchées.

Elles produisent généralement des solutions de bonne qualité sans garantir l'optimalité pour des problèmes de grande taille.

Elles peuvent se diviser en trois classes distinctes :

- les heuristiques de construction ;

- les heuristiques à deux phases ;
- les heuristiques d'amélioration.

Parmi les heuristiques qui traitent le VRP, nous citons :

- **L'heuristique "groupe en premier, route en second" :**

C'est une des heuristiques les plus connues. Elle se base sur l'aspect géométrique du problème. Elle consiste à créer des groupes de clients qui sont géographiquement voisins puis à construire les tournées à l'intérieur de ce dernier.

- **L'heuristique "route en premier, groupe en second" :**

Le principe de cette heuristique est de construire des tournées comportant un grand nombre de clients, qui sont réellement non réalisables, puis de les subdiviser en de petites tournées pour obtenir des solutions réalisables pour le VRP.

✓ **Les métaheuristiques :** Les méta-heuristiques sont des stratégies d'exploration du domaine de solutions réalisables à la recherche de la solution optimale ou presque. Les méthodes basées sur les méta-heuristiques sont à ce jour des algorithmes spécialisés et qui ont les caractéristiques suivantes :

- **Simplicité :** les métaheuristiques reposent sur un principe simple et intuitif ;
- **Généralités :** puisqu'elles sont applicables à une large gamme de problèmes d'optimisation ;
- **Absence de garantie de performance :** puisqu'elles sont des procédures qui ne garantissent pas d'obtenir l'optimum ni de s'en approcher, elles sont destinées essentiellement à la résolution de problèmes de grande taille.

Dans la littérature, les métaheuristiques sont subdivisées en deux grandes familles :

- Les métaheuristiques de recherche locale :
 - le recuit Simulé ;
 - la méthode de la descente ;
 - la Recherche Tabou.
- Les métaheuristiques d'évolution :

- les algorithmes génétiques ;
- la recherche distribuée ;
- les colonies de fourmis ;

Conclusion

Dans ce chapitre, on a présenté le principe général du problème de tournées de véhicules (VRP) qui appartient à la classe des problèmes NP-DURS, sa modélisation mathématique, les différentes contraintes intervenant dans sa classification, un aperçu général des méthodes utilisées pour sa résolution.

Quand les véhicules sont des bus scolaires qui transportent des élèves, là on se ramène au cas du problème SBRP (School Bus Routing Problem) qui sera traité dans le prochain chapitre.

2

Problème de tournées des autobus scolaires (SBRP)

Contents

Introduction	15
2.1 Revue de la littérature	16
2.2 Modélisation du problème	19
Conclusion	22

Introduction

Le problème de tournées des autobus scolaires (School Bus Routing Problem, SBRP) est un problème courant de la vie réelle proposé dans la littérature par Newton et Thomas (1969)[20]. Le SBRP a les mêmes caractéristiques avec le problème de tournées de véhicules (VRP), hors que le VRP traite le transport de marchandises, et le SBRP est lié aux transports des élèves.

Le SBRP général cherche à planifier des itinéraires optimaux pour une flotte d'autobus scolaires, chaque autobus prenant en charge les élèves à différents arrêts de bus, puis les conduisant à l'école. Diverses contraintes doivent être satisfaites : capacité maximale des bus, heure de livraison ou créneau horaire à l'école. Dans ce chapitre, nous allons introduire une synthèse bibliographique sur les travaux qui ont été réalisés sur le SBRP, puis on va modéliser le problème dans le cas de transport universitaire.

2.1 Revue de la littérature

Différentes méthodes, techniques et modèles mathématiques ont été développés pour traiter le problème de l'acheminement des bus scolaires, également appelé SBRP. Toutefois, il reste nécessaire d'adopter une approche générale du SBRP, car la plupart des études sur ce sujet ont été réalisées en raison de l'apparition de problèmes du monde réel.

Cette section vise à fournir une revue de problèmes de routage des autobus scolaires (SBRP).

Spada et al. (2005) ont examiné un problème d'acheminement des élèves de leurs lieux de résidence vers différentes écoles. Le modèle met l'accent sur l'optimisation du niveau de service du système de bus scolaires et vise à minimiser les temps de séjour (temps d'attente à l'arrêt plus le temps passé dans le bus) de chaque élève, ce qui est modélisé par deux fonctions objectives différentes. Les écoles sont considérées par ordre croissant d'heures de départ. Cependant, ils ont obtenu un problème de programmation entière non linéaire. Par conséquent, ils ont proposé et analysé des heuristiques pour construire une solution. Ces heuristiques commencent toutes par générer une solution initiale réalisable par une méthode gourmande qui prend en compte la capacité minimale du véhicule. Dans la plupart des cas, la solution n'est pas satisfaisante et doit être améliorée. Ils ont proposé donc trois méta-heuristiques pour améliorer la qualité de la solution : la recherche tabou et deux variantes du recuit simulé.

Ces méta-heuristiques ont été testées sur 30 instances chacune. Les résultats obtenus montrent que le recuit simulé est la plus efficace des trois, quel que soit le choix des paramètres [22].

Enfin, ils ont donné un aperçu d'un outil d'aide à la décision interactif. Son objectif principal est d'analyser la pertinence de la solution produite par l'heuristique et éventuellement de la modifier manuellement. Ceci est particulièrement important lorsque les conditions pour le fonctionnement du bus sont modifiées. En effet, dans ce cas, il est préférable de créer manuellement une nouvelle solution à partir de la solution d'origine, au lieu de relancer l'ensemble du processus sur le problème modifié.

Byung-In Kim , Seongbae Kim , Junhyuk Park en (2012) ont introduit un problème d'optimisation des horaires des bus, pour desservir tous les trajets donnés en tenant compte des créneaux horaires des écoles. L'objectif principal retenu par les auteurs est la minimisation du nombre de bus, ainsi que de la distance totale parcourue. Le problème au début a été traité comme un VRPTW. Les deux approches exactes basées sur un problème d'affectation ont ensuite été proposées pour des cas particuliers et un algorithme heuristique basé sur un problème d'affectation a été proposé pour des cas plus généraux. Des problèmes de référence ont été développés. Les expériences de calcul ont montré l'exactitude et l'effica-

citée des algorithmes. En résumé, les modèles MIP et l'algorithme de branch and bound basé sur les problèmes d'affectation peuvent être utilisés pour les petits problèmes de flotte homogènes et l'algorithme heuristique pour les grands problèmes et les problèmes hétérogènes[16].

Des auteurs tels que Khalid et al, en 2012 ont présenté la modélisation du problème de tournées des bus scolaires (SBRP) qui a été formulé en tant que SDVRP, en tenant compte des capacités et des créneaux horaires des bus dans l'école dans laquelle le bus fonctionne, afin d'obtenir les informations à utiliser dans le modèle, des outils du système d'information géographique (GIS) pour gérer les données spatiales, effectuer les opérations d'analyse spatiale requises et visualiser les résultats, et des techniques de regroupement où elles trient les emplacements de collecte en fonction de leur emplacement vers l'emplacement de destination, puis il regroupe les emplacements de collecte en groupes en fonction de la capacité du bus, et des techniques de découpage de réseau ont été utilisés, tel que l'objectif principal de l'algorithme de découpage de réseau proposé est de découper l'ensemble du réseau recherché en réseaux personnalisés en fonction des limites de chaque cluster. Afin de limiter l'espace de recherche. Tout d'abord, tout le réseau et tous les clusters sont chargés. Ensuite, pour chaque cluster, l'algorithme obtient les limites X et Y. Ensuite, l'algorithme coupe l'ensemble du réseau en fonction de ces limites X et Y, de sorte que chaque cluster dispose d'un réseau spécifique et limité. Enfin, les réseaux de ces clusters sont enregistrés et insérés dans le module de routage, où l'on tente de trouver une solution au modèle (SBRP), en utilisant les métaheuristiques hybrides d'optimisation des colonies de fourmis (ACO) avec l'heuristique d'amélioration locale, Lin Kernighan itérée. En traitant le cas de L'école ABC qui est située dans la ville de Zagazig, du gouvernorat de Sharqia, Égypte. L'école ABC compte 120 élèves répartis dans toute la ville de Zagazig et la capacité de chaque bus est de 28 élèves. L'étude de cas a permis d'avoir un regroupement en 5 clusters avec 5 itinéraires différents pour le ramassage de ces élèves, ce qui prouve que les résultats rapportés du cadre proposé surpassent généralement ceux du solveur de routage ArcGIS Network Analyst[8].

En 2014, E GurpreetSingh et V Dhir ont présenté une approche pour résoudre le problème de tournées des véhicules ouvert (Open Vehicle Routing Problem) pour les autobus scolaires basée sur un algorithme de colonie de fourmis. L'étude a été faite à l'école internationale Sant Baba Bhag Singh située dans une zone rurale de Padhiana, au Pendjab, en Inde, où plus de 1 000 étudiants sont transportés par bus qui ont une capacité limitée, à partir de différents villages vers leurs écoles en parcourant un trajet loin, le problème principal est que certains bus ne retournent pas à l'école c'est le cas de l'OVRP qui diffère du problème bien connu de tournées de véhicules (VRP) puisque dans ce dernier, les bus doivent suivre le même chemin dans l'ordre inverse.

L'objectif de l'étude de ce problème de tournées est de sélectionner un sous-ensemble d'arrêts qui seront effectivement visités par les bus, et d'élaborer un ensemble de circuits qui minimisent la distance totale parcourue par tous les bus. Pour atteindre cet objectif l'algorithme (ACO) a été proposé, et il comprend deux parties : la première partie introduit l'algorithme ACO avec élimination de boucle de tour. Une comparaison entre OVRP et

VRP a été décrite dans la deuxième partie afin de déterminer le meilleur itinéraire pour les bus et la livraison aux étudiants à temps.

Après avoir codé l'algorithme dans MATLAB-2011b, les résultats obtenus montrent que pour une grande distance, l'itinéraire ouvert est préférable par rapport à un itinéraire fermé (avec retour au dépôt)[11].

En raison de la fréquence des accidents des bus scolaires, au cours desquels de nombreux élèves ont été blessés, voire tués. Par exemple, en Chine, Les autorités en charge de la sécurité ont trouvé après avoir fait une enquête, qu'en raison du budget limité, l'offre actuelle d'autobus scolaires pour de nombreuses écoles est insuffisante pour couvrir toute la demande outre le problème de la surcharge, en raison de la répartition dispersée des élèves dans certaines zones, certains bus scolaires sont presque vides. La raison pour laquelle, B Yao, Q Cao, Z Wang, P Hu, M Zhang et B Yu, en 2016, ont proposé un algorithme heuristique à deux étapes pour le problème de tournées des bus scolaires avec un plan de chargement mixte (school bus routing problem with mixed load plan, SBRP-MLP) qui est progressivement appliqué au fonctionnement des autobus scolaires. Cependant, SBRP-MLP est difficile à résoudre par les algorithmes heuristiques traditionnels en raison de sa complexité inhérente.

Contrairement au SBRP avec plan de chargement mixte où un autobus scolaire dans SBRP-MLP peut visiter plusieurs écoles, la solution de SBRP avec un plan de charge unique conduirait généralement à une faible utilisation de la capacité des bus scolaires et à un coût de fonctionnement élevé. Du fait, cet article conçoit un algorithme heuristique à deux étapes pour SBRP-MLP. Lors de la première étape, un algorithme de clusterisation basé sur l'agrégation est introduit pour regrouper les arrêts ayant des tendances similaires dans le même cluster d'arrêt. À la deuxième étape, un algorithme de colonie de fourmis (Ant Colony Optimisation, ACO) amélioré est proposé pour optimiser les itinéraires des bus scolaires pour la visite des arrêts dans chaque groupe d'arrêts, tout en proposant deux modes de fonctionnement de bus scolaire pour SBRP-MLP : le mode SBRP-MLP avec arrêts virtuels et le mode SBRP-MLP avec transport interscholastique. Enfin, plusieurs instances de tests sont construites pour comparer les résultats d'optimisation pour SBRP-MLP et SBRP-SLP, ainsi que les performances de l'algorithme proposé.

Les résultats ont montré que le mode SBRP-MLP avec arrêts virtuels pourrait être meilleur pour les cas de petite taille, tandis que le mode SBRP-MLP avec transport interscolaire convient aux cas de grande taille. La contrainte de la fenêtre temporelle des écoles est ignorée et l'optimisation de l'itinéraire des autobus scolaires est supposée ne concerner que la distance parcourue par les autobus scolaires[25].

En Juin 2018, Israel Herrera, Leandro Leonardo Lorente Leyva, Diego Peluffo et Robert Valencia-Chapi ont présenté une revue sur l'optimisation du transport universitaire étudiée dans la ville d'Ibarra, Equador. Dans le but de :

- faciliter la mobilisation ;
- améliorer le service de transport des étudiants ;
- optimiser le temps de transfert ;

- optimiser l'utilisation des unités du transport disponible.

Pour ce faire, les 314 étudiants de la ville d'Ibarra ont été placés sur une carte à l'échelle, et puisque l'itinéraire initial emprunté par les autobus de transport universitaire de destination appelé "stade UTN" et le retour à l'université Técnica del Norte offre un faible niveau de service aux étudiants, environ 22,61 %, avec une distance de 8,2 km en 18 minutes et le temps total pour terminer le tour était de 30 minutes, ils ont pris un échantillon de 50 noeuds d'étudiants parmi les 314 étudiants en passant directement chez eux, ensuite ils ont obtenu un itinéraire long et coûteux (56.8 km en 2h 8min), et cette durée augmenterait de façon exponentielle si l'on tente de ramasser tous les 314 étudiants de chez eux. Afin de faciliter le problème, il est nécessaire de faire plusieurs partitions de la population étudiante, à l'aide de l'algorithme de clustering K-means, puis les évaluer selon certains critères d'arrêts de manière à rendre possible l'acheminement des véhicules.

Après avoir effectué un regroupement des étudiants, les itinéraires initiaux ont été déterminés, si ces itinéraires peuvent être améliorés, des installations sont localisées géographiquement au moyen de l'enquête adressée aux étudiants.

Ensuite, l'application OSRM (Open Source Routing Machine) (plate-forme gratuite liée à l'algorithme CH) simule l'itinéraire optimal, si cet itinéraire ne répond pas aux exigences des étudiants donc on refait la localisation géographique des installations à l'aide de l'algorithme K-means, et on simule à nouveau l'itinéraire à l'aide de la méthode CH.

Enfin, l'amélioration est déterminée au moyen de l'indicateur de proximité d'un service de transport.

Il a été déterminé que l'exécution avec 4 arrêts (K-centroïdes) et les Trois installations fixes (FICA, Stade UTN et ancien hôpital San Vicente de Paul) est l'optimum.

Une comparaison entre l'itinéraire initial déjà cité et l'itinéraire optimal a été faite, dont l'indicateur de proximité des arrêts de transport sur l'itinéraire initial était de 22,61%, ce qui entraîne un faible niveau de service du transport proposé. Cet indicateur est amélioré avec la route optimale proposée, qui offre un niveau de couverture de 88% en établissant un trajet de 16.3 km en 33 minutes [12].

2.2 Modélisation du problème

La modélisation d'un problème donné est une étape très importante qui consiste à faire une conception du modèle afin de pouvoir l'optimiser. Un problème moitié modélisé, moitié résolu, et pour bien modéliser notre problème, nous allons suivre les étapes suivantes :

- **Formulation du problème** : dans cette première étape, le problème est identifié et déclaré. Ensuite, la détermination de le ou les objectifs ;
- **Construction du modèle** : Dans cette étape importante, un modèle mathématique

abstrait est construit pour le problème, d'où on peut s'inspirer des modèles similaires de la littérature. Habituellement, les modèles qu'on résout sont des simplifications de la réalité. Parfois, ils omettent des processus complexes à représenter dans un modèle mathématique, et cela ce fait en introduisant une fonction objectif limité par des contraintes.

2.2.1 Modèle

Nous avons défini un réseau $G = (V, E)$ où V est l'ensemble de tous les arrêts (arrêts campus et arrêts maison) qu'on appelle aussi sommets, et E l'ensemble des arêtes. On note t_{ij}^k le temps mis par le véhicule k pour aller du sommet v_i au sommet v_j .

Pour transporter les étudiants, m bus sont disponibles tel que ces bus sont de capacité homogène.

Soient les notations suivantes :

- S l'ensemble d'arrêts campus ;
- C l'ensemble des groupes d'étudiants ;
- c groupe d'étudiants,
- n nombre de clients (ou sommets) ;
- m nombre de véhicules $m \leq n$;
- d_{ij} distance entre le sommet v_i et le sommet v_j , si $d_{ij} = \infty$ alors le chemin entre i et j n'existe pas (impasse, rue piétonne, ...);
- t_i^k l'instant d'arrêt du véhicule k au sommet v_i ;
- T_{max}^K le temps maximal de la tournée du véhicule k ;
- Q^k capacité du bus k ;
- $i = 0, \dots, n$ est l'indice des sommets prédécesseurs ;
- $j = 0, \dots, n$ est l'indice des sommets successeurs, tel que 0 indique le dépôt ;
- $k = 1, \dots, m$ est l'indice des véhicules ;
- Les variables de décision du problème sont :

$$E_i^k = \begin{cases} 1 & \text{si le bus } k \text{ est éligible pour desservir l'arrêt } v_i; \\ 0 & \text{sinon.} \end{cases} \quad (2.1)$$

$$x_{ij}^k = \begin{cases} 1 & \text{si } (v_i, v_j) \text{ est parcouru par le véhicule } k; \\ 0 & \text{sinon.} \end{cases} \quad (2.2)$$

- O_s : l'instant de début de cours de l'école $s(c)$;
- τ_s : l'instant d'arrivé à l'école $s(c)$;
- w_i^k : l'heure de début de service de l'arrêt v_i par le bus k ;
- $[a_i, b_i]$: la fenêtre temporelle de début de service à l'arrêt v_i .

La fonction à minimiser est donnée comme suit :

$$\min \sum_{i=0}^m \sum_{j=0}^n t_{ij} \sum_{k=1}^n x_{ij}^k. \quad (2.3)$$

Sous les contraintes :

$$\sum_{k=1}^m \sum_{i=0}^n x_{ij}^k \geq 1; \quad \forall j \in V'; \quad (2.4)$$

$$\sum_{k=1}^m \sum_{i=0}^n x_{ij}^k \geq 1; \quad \forall i \in V'; \quad (2.5)$$

$$\sum_{i=0}^n \sum_{l=0}^n x_{il}^k = \sum_{l=0}^n \sum_{j=1}^n x_{lj}^k; \quad \forall 1 \leq i \leq n; \quad (2.6)$$

$$t_j^k - t_i^k \geq t_{ij}^k; \quad \forall i = 0, \dots, n, \quad \forall j = 0, \dots, n, \quad \forall k = 1, \dots, m; \quad (2.7)$$

$$O_s \leq \tau_s; \quad \forall s \in S; \quad (2.8)$$

$$\sum_{i=0}^n t_i^k \sum_{k=1}^m \sum_{j=0}^n x_{ij}^k + \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^m t_{ij}^k x_{ij}^k \leq T_{max}^k; \quad (2.9)$$

$$a_j \sum_{i \in \delta^-(j)} \sum_{k=1}^m x_{ij}^k \leq w_i^k \leq b_j \sum_{i \in \delta^-(j)} \sum_{k=1}^m x_{ij}^k; \quad \forall j = 0, \dots, n; \quad (2.10)$$

$$\sum_{k=1}^m \sum_{i \in \delta^-(j)} x_{ij}^k = 1; \quad \forall j = 0, \dots, n; \quad (2.11)$$

$$\sum_{i=0}^n q_i \left(\sum_{j=0}^n x_{ij}^k \right) \leq Q^k; \quad \forall k = 1, \dots, m; \quad (2.12)$$

$$\sum_{i \in S} \sum_{\substack{j \in S \\ j \neq i}} x_{ij}^k \leq |S| - 1; \quad \forall S \subset V', |S| \geq 2; \forall k = 1, \dots, m; \quad (2.13)$$

$$E_i^k \in \{0, 1\}; \quad x_{ij}^k \in \{0, 1\}; \quad \forall 0 \leq (i, j) \leq n; \quad 1 \leq k \leq m. \quad (2.14)$$

La fonction-objectif (1.3) cherche à minimiser le temps de toutes les tournées. Les contraintes (2.4) et (2.5) garantissent qu'un arrêt peut être servi au moins une fois par un ou plusieurs véhicules. La contrainte (2.6) assure la conservation du flux et la continuité de la tournée. Pour chaque tournée, l'horaire doit être tel que le bus puisse se déplacer tout en respectant le temps de trajet minimum, ce qui est montré dans la contrainte (2.7). De plus, chaque étudiant doit arrivé au campus à l'heure (2.8). La contrainte (2.9) permet d'assurer que la

durée totale de la tournée ne dépasse pas sa durée totale maximale. La contrainte (2.10) est la contrainte de fenêtre temporelle. La contrainte (2.11) garantit qu'un bus éligible doit être affecté à chaque arrêt. Les bus ne peuvent transporter plus d'étudiants que leurs capacités ne leur ont pas permis, en chaque arrêt de chaque bus, la contrainte (2.12) doit donc être vérifiée. La contrainte (2.13) assure l'empêchement des sous-tournées. La contrainte (2.14) est la contrainte de binarité des variables de décisions x_{ij}^k et E_i^k .

Conclusion

Dans ce chapitre, nous avons décrit le problème réel de routage des autobus scolaires (SBRP) et introduit une revue de la littérature sur ce problème en fonction de ses divers aspects. Puis, le problème a été modélisé dans le cas de transport universitaire sous forme d'un programme linéaire.

Vu la complexité de la résolution du SBRP, nous avons opté pour l'utilisation de la métaheuristique colonie de fourmis (ACO), qui va être définie dans le prochain chapitre.

3

Approches de Résolution

Contents

Introduction	23
3.1 Approches exactes	24
3.2 Approches heuristiques	28
3.3 Méta-heuristiques	32
Conclusion	40

Introduction

Suivant la complexité du problème des tournées de véhicules, on distingue essentiellement deux catégories de méthodes pour la résolution du VRP, et de ses variantes qui est du type NP-difficile : les méthodes exactes et les méthodes de résolution heuristiques et méta-heuristiques, qui seront introduites dans ce chapitre particulièrement la méta-heuristique de colonie de fourmis.

Les méthodes exactes obtiennent des solutions optimales et garantissent leur optimalité, les méthodes approximatives (ou heuristiques, méta-heuristiques) génèrent des solutions de haute qualité dans un délai raisonnable pour une utilisation pratique, mais il n'y a aucune garantie de trouver une solution globale optimale.

3.1 Approches exactes

Le principe général des méthodes exactes pour la résolution des problèmes de l'optimisation combinatoire consiste à énumérer implicitement et le plus efficacement possible, toutes les solutions du problème pour en extraire une solution garantie optimale, en un temps de calcul, le plus court possible. Bien que les problèmes réalistes de VRP soient souvent difficiles à résoudre, des méthodes exactes ont été utilisées avec un certain succès, pour des instances non volumineuses.

Dans la classe des méthodes exactes, on trouve les algorithmes classiques suivants : programmation dynamique, famille d'algorithmes branches et X (branch and bound, branch and cut, branch and price) développés dans la communauté de recherche opérationnelle. Ces méthodes énumératives peuvent être considérées comme des algorithmes de recherche arborescente.

3.1.1 Programmation Linéaire

Une technique des mathématiques est connue sous le nom de programmation linéaire, elle permet de déterminer la meilleure solution d'un problème tout en satisfaisant une série d'équations et d'inéquations linéaires.

Un programme linéaire est un problème d'optimisation consistant à maximiser (ou minimiser) une fonction objectif linéaire de n variables de décision soumises à un ensemble de contraintes.

Tous les programmes linéaires comportent trois éléments capitaux :

- ✓ Fonction objectif : elle sert de critères pour déterminer la meilleure solution à un problème d'optimisation ;
- ✓ Les contraintes : les conditions qui peuvent limiter les valeurs que peuvent prendre les variables de décision ;
- ✓ Variables de décision : elles décrivent les décisions à prendre.

Programme linéaire

Un problème de programmation linéaire (**PL**) est un problème d'optimisation où la fonction objectif à plusieurs variables et les contraintes sont toutes linéaires, sa forme générale est la suivante :

$$\left\| \begin{array}{l} \max \text{ (ou min) } \quad Z = \sum_{j=1}^n c_j x_j; \\ \text{s.c.} \quad \quad \quad \sum_{j=1}^n a_{ij} x_j \leq, =, \geq b_i; \quad \forall i = 1, \dots, m; \\ \quad \quad \quad x_j \geq 0; \quad \quad \quad \forall j = 1, \dots, n. \end{array} \right. \quad (3.1)$$

tel que :

- n : nombre de variables de décision (x_1, \dots, x_n) du problème ;
- m : nombre de contraintes réelles , avec $m \leq n$;
- (c_1, \dots, c_n) : vecteur ligne des profits (ou gains) ;
- $\{(a_{ij}); i = 1, \dots, m; j = 1, \dots, n\}$ sont des constantes réelles ;
- (b_1, \dots, b_m) : vecteur colonne des seconds membres.

Pour résoudre un programme linéaire, on utilise les méthodes suivantes : la méthode graphique, l'algorithme de simplexe.

Le terme programmation linéaire suppose que les solutions à trouver doivent être représentées en variables réelles. S'il est nécessaire d'utiliser des variables discrètes dans la modélisation du problème, on parle alors de programmation linéaire en nombres entiers (PLNE), représenté comme suit :

$$\left\| \begin{array}{ll} \text{Optimiser} & Z = cx; \\ \text{s.c} & Ax \leq, =, \geq b; \\ & x_j \in \mathcal{N}; \quad \forall j = 1, \dots, n. \end{array} \right. \quad (3.2)$$

On dit qu'un programme linéaire en $\{0, 1\}$ ou un programme linéaire en variable bivalentes, si les variables sont binaires $x \in \{0, 1\}$.

3.1.2 Programmation dynamique

La programmation dynamique est basée sur la division récursive d'un problème en sous-problèmes plus simples. Cette procédure est basée sur le principe de Richard Bellman (1957), selon lequel "la sous-politique d'une politique optimale est elle-même optimale" [4]. Cette méthode d'optimisation par étapes est le résultat d'une séquence de décisions partielles. La procédure évite une énumération totale de l'espace de recherche en élaguant des séquences de décision partielles qui ne peuvent pas conduire à la solution optimale.

- Définir les étapes et les états. Un problème peut être divisé en plusieurs étapes N . Un certain nombre d'états sont associés à chaque étape.
- Définir le coût de la phase initiale et des états. Il existe un état initial du système x_0 .
- Définir la relation récursive pour un état à l'étape k en termes d'états des étapes précédentes. Le système prend l'état x_k à l'étape k . Au stade k , l'état du système passe de x_k à x_{k+1} .

Eilon, Wastson-Gandy et Christofides (1971) ont appliqué ce principe pour la résolution du SVRP. Ils l'ont subdivisé en sous-problèmes simples et faciles à résoudre. Les sous problèmes obtenus sont de taille réduite et peuvent être traités. Ils sont liés au problème principal par des contraintes supplémentaires. Leur résolution séquentielle suivie d'une remontée graduelle au problème principal tout en commençant par les sous-problèmes les

plus petits et on remonte vers les sous-problèmes de plus en plus difficiles en tirant profit des résultats des problèmes déjà obtenus pour que les calculs à exécuter restent simples et rapides, ce qui assure la détermination de la solution du problème principal.

3.1.3 Approche de Branch and X

A. Branch and bound :

L'algorithme de séparation et d'évaluation, est l'une des méthodes les plus courantes pour résoudre les problèmes d'optimisation de manière exacte. L'algorithme est basé sur une énumération implicite de toutes les solutions du problème d'optimisation considéré. L'espace de recherche est exploré en créant dynamiquement un arbre dont le nœud racine représente le problème en cours de résolution et l'ensemble de son espace de recherche associé. Les nœuds feuille sont les solutions potentielles et les nœuds internes sont des sous-problèmes de l'espace total de la solution. La taille des sous-problèmes est de plus en plus réduite à l'approche des feuilles.

La construction d'un tel arbre et son exploration sont effectuées à l'aide de deux opérateurs principaux : la ramification et l'élagage.

L'algorithme procède en plusieurs itérations au cours desquelles la solution la mieux trouvée est progressivement améliorée. Les nœuds générés et non encore traités sont conservés dans une liste dont le contenu initial est limité au nœud racine. Les deux opérateurs interviennent à chaque itération de l'algorithme. La stratégie de branchement détermine l'ordre dans lequel les branches sont explorées. La stratégie d'élagage élimine les solutions partielles qui ne conduisent pas à des solutions optimales. Cela se fait en calculant la limite inférieure associée à une solution partielle. Si la limite inférieure d'un nœud (solution partielle) est supérieure à la meilleure solution trouvée jusqu'à présent ou à une limite supérieure connue du problème, l'exploration du nœud n'est pas nécessaire. L'algorithme se termine s'il n'y a plus de nœuds à brancher ou si tous les nœuds sont éliminés. Par conséquent, les concepts les plus importants dans la conception d'un algorithme efficace de branch and bound sont la qualité des limites et la stratégie de ramification.

Christofied, Mingozzi et Toth ont appliqué la méthode de séparation et évaluation au VRPTW-larges à contraintes de capacités de véhicules variables. La meilleure instance résolue est de 53 clients et 08 véhicules. Après résolution du problème, le nombre maximal de véhicules dans une tournée est de 15.

B. Branch and Cut and Price :

Les techniques suivantes ont pour but de générer des relaxations plus strictes au programme linéaire.

Plan de coupe :

Des approches de plan de coupe ont été proposées en 1958 par Gomory. L'utilisation de coupes peut grandement améliorer les algorithmes de branchement. En général, les algorithmes de plan de coupe consistent à ajouter de manière itérative des contraintes spécifiques à la relaxation du problème de PL. Ces contraintes représentent des restrictions au problème de sorte que le polytope de programmation linéaire se rapproche étroitement du polyèdre représenté par la coque convexe de toutes les solutions possibles du problème.

Génération de colonnes : (décomposition de Dantzig – Wolfe, par exemple)

Elle génère une décomposition du problème en maître et en sous-problèmes [24]. Elle a été appliquée pour résoudre le VRP avec des temps de parcours stochastiques (Kenyon et Morton 2003). Cette méthode est applicable lorsque l'échantillon n'est pas grand et qu'il est proposé de traiter les deux versions du problème : la minimisation du temps d'achèvement et la maximisation de la probabilité d'être achevé dans un délai cible. Cependant, les tests ont été effectués uniquement pour le premier cas. Une probabilité d'occurrence est attribuée à chacune des réalisations des paramètres stochastiques. Le problème est ensuite modélisé comme la minimisation de la valeur attendue de la fonction objectif. Pour chaque réalisation possible des paramètres, un problème déterministe est résolu. La méthode est basée sur les branches et est découpée comme suit. Les contraintes d'élimination de sous-traitants sont assouplies. Le problème d'optimisation est résolu et s'il n'existe pas de sous-tour, la solution est optimale. S'il existe des sous-tours n'incluant pas le dépôt, une nouvelle solution est construite, reliant chaque sous-tour au tour principal attribué au même véhicule. La nouvelle solution est évaluée et si elle se situe dans un pourcentage présélectionné de la solution optimale, l'algorithme s'arrête. Le pourcentage est calculé en utilisant la fonction objectif de la solution au problème relâché (borne inférieure) comme référence. Si la nouvelle solution ne se situe pas dans le pourcentage donné de la solution optimale, des contraintes d'élimination de sous-traités sont ajoutées et la procédure est répétée. La méthode est testée sur quatre instances, représentant chacune un réseau de neuf nœuds et disposant d'un parc de deux véhicules. Le temps de trajet suit une distribution discrète. Les résultats sont comparés aux solutions optimales au problème, dans lesquelles les valeurs moyennes des paramètres sont utilisées. Les solutions aux modèles stochastiques (temps de réalisation) étaient meilleures [19].

Une décomposition similaire est utilisée pour reformuler le CVRP robuste avec des délais et une incertitude sur le temps de déplacement / la demande (Lee et al. 2012). Le problème est initialement formulé comme un ensemble couvrant le chemin, dans lequel la variable de décision consiste à inclure ou non un itinéraire dans la solution à l'aide d'un algorithme de Branch and Price. Les contraintes d'intégrité sont alors assouplies et le problème est résolu avec un ensemble restreint de routes réalisables, car le nombre total de routes possibles peut être très important. Dans ce contexte,

un itinéraire réalisable est celui qui respecte les délais et les contraintes de capacité de chaque client, tandis que la plupart des paramètres incertains présentent des déviations maximales. Le sous-problème de génération de colonne est résolu pour trouver une colonne avec un coût réduit négatif et si aucune colonne n'est trouvée, la procédure est terminée. Un algorithme d'étiquetage (programmation dynamique) est utilisé pour rechercher des itinéraires réalisables avec un coût réduit négatif. L'existence d'une solution entière optimale sans cycles n'est pas garantie. Si la solution optimale a des cycles, une procédure de branchement est appliquée. Les solutions obtenues par le modèle robuste sont comparées aux solutions obtenues par un CVRP déterministe équivalent. Un ensemble de scénarios est généré et les tests estiment dans quel pourcentage de scénarios la faisabilité de la solution est conservée (robustesse). Dans trois ensembles différents d'instances, le modèle robuste améliore la robustesse d'environ 48%, 82% et 64%, en moyenne, par rapport à l'approche déterministe[19].

3.2 Approches heuristiques

Les heuristiques constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grandes tailles lorsqu'on peut se contenter d'une solution approximative. Selon le processus de génération de solutions, nous distinguons principalement trois types d'heuristiques : les procédures de construction de tours et les procédures d'amélioration de tours, une troisième classe de méthodes dites "à deux phases" vient s'ajouter à ces deux classes et reflète la décomposition du VRP en deux sous-problèmes : l'affectation des clients aux véhicules et l'optimisation des tournées de ces derniers.

3.2.1 Heuristiques de construction

Les méthodes constructives sont des méthodes itératives où, à chaque itération, une solution partielle est complétée. Elles construisent une route en choisissant les arcs à inclure d'une façon séquentielle. Par la suite, on améliore cette solution initiale en la comparant avec une nouvelle afin de produire une économie (Golden, 1984). On peut citer quelques algorithmes, tels que :

1. **Algorithme de gain :**

Une des premières heuristiques constructives pour le VRP était l'algorithme de gain de Clarke et Wright ou la méthode des économies (savings) proposée en (1964)[1] qui commence par une solution initiale composée de tournées aller-retour c-à-d créer une tournée pour chaque client. Puis itérativement fusionne les tournées jusqu'à ce que la distance totale ne puisse plus être réduite. Comme avec cet exemple illustré dans la Figure (3.1) :

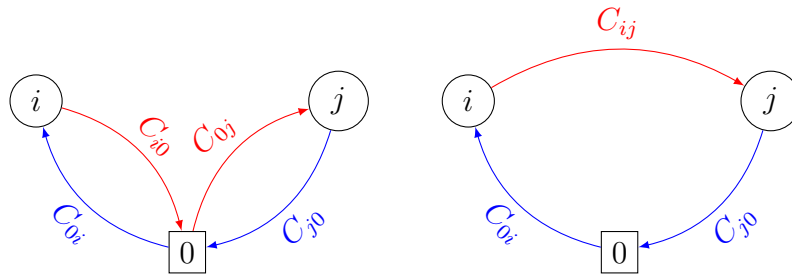


FIGURE 3.1 – Illustration du principe de l'algorithme de gain

où on fusionne une tournée qui se termine par l'emplacement i avec une autre tournée qui commence par l'emplacement j , en maximisant le gain S_{ij} , tel que $S_{ij} = c_{i0} + c_{0j} - c_{ij}$ et c_{ij} est le coût de déplacement de l'emplacement i vers l'emplacement j , où l'emplacement 0 est le dépôt. Ce processus s'arrête lorsque ne reste aucune tournée faisable puisse être fusionné.

2. Algorithme de balayage :

L'algorithme de balayage de Gillett et Miller (1974) est considéré comme l'un des meilleurs algorithmes de construction [18]. L'algorithme de balayage comporte deux principales étapes : la construction d'une solution initiale et sa post-optimisation. Tout d'abord, on numérote les clients dans l'ordre croissant de leurs distances par rapport au dépôt. La création de la solution initiale se fait en partant du premier client. Des clients sont ensuite ajoutés à la tournée dans l'ordre croissant de leur numéro tant que les contraintes de capacité du véhicule et de distance parcourue sont respectées. Lorsqu'une tournée est terminée, on en recommence une autre en débutant avec le numéro suivant et ce jusqu'à ce que tous les clients soient desservis. Ensuite, pour la post-optimisation, des permutations entre les clients des tournées sont faites afin de déterminer s'il est possible de réduire la distance totale parcourue. Le processus de construction de tournées sera ensuite répété en commençant par différents clients et aussi en construisant les tournées en sens décroissant des numéros de clients et la solution avec la distance parcourue la plus petite sera conservée.

3. Algorithme de pétale :

Les algorithmes pétales proposés par Laporte et Renaud (1996), sont une extension des algorithmes de balayage : les tournées générées (pétales) sont sélectionnées en résolvant un problème de partitionnement. Soit l'algorithme de pétale :

$$\min \sum_{k \in S} d_k x_k. \quad (3.3)$$

Sous :

$$\sum_{k \in S} a_{ik} x_k = 1; \quad i = 1, \dots, n; \quad (3.4)$$

$$x_k = 0 \text{ ou } 1; \quad k \in S; \quad (3.5)$$

où : S est l'ensemble des itinéraires, $x_k = 1$ si et seulement si l'itinéraire k appartient à la solution, a_{ik} est le paramètre binaire et n'est égal à 1 que si le sommet i appartient à l'itinéraire k , et d_k est le coût de pétale k .

3.2.2 Méthodes à deux phases

Les méthodes à deux phases sont basées sur la décomposition du VRP en deux sous-problèmes : le sous-problème de partitionnement des clients en sous-groupes et le sous-problème de détermination des routes pour chaque sous-groupe. Elles se décomposent naturellement en deux classes suivant l'ordre dans lequel les deux phases (ou sous-problèmes) sont traitées.

Méthode " *Cluster first - Route second* "

Elle consiste à regrouper les clients en groupes sans violer la capacité du véhicule et sans spécifier l'ordre de la tournée, puis à résoudre un TSP pour chaque groupe.

L'algorithme SWEEP commence d'abord par attribuer aux clients des coordonnées angulaires relatives au dépôt, puis effectue un balayage dans le sens des coordonnées croissantes. Les clients sont ainsi ajoutés successivement, et dès qu'une des contraintes du problème est violée, un nouveau groupe est créé. Une fois les groupes formés, l'algorithme procède à la résolution d'un TSP sur chacun d'entre eux[17].

Méthode " *Route first - Cluster second* "

Elle commence par la conception d'une tournée géante en visitant tous les clients, et qui est ensuite divisée en de petites tournées faisables.

3.2.3 Heuristiques d'amélioration

Elles nécessitent une solution de départ produite par une heuristique constructive, qui s'améliore aux cours du déroulement de l'algorithme, parmi ces heuristiques, on cite la recherche locale qui a pour but de trouver les meilleures solutions dans le voisinage d'une solution donnée, en effectuant des transformations ou des mouvements qui réduisent le coût total. Lorsqu'une meilleure solution est obtenue, elle remplace l'ancienne, et elle s'arrête dès qu'il n'existe plus de mouvement améliorant. Il existe plusieurs types d'opérateurs de mouvement, les mouvements qui consistent à améliorer chaque tournée séparément, comme le mouvement k-opt, et Or-opt.

Le principe d'un mouvement k-opt est d'enlever k arêtes non consécutives d'une tournée et les remplacer par k autres arêtes de manière à constituer une nouvelle solution réalisable pour le problème traité. En général $k = 2$ ou $k = 3$, car l'exploration du voisinage k-opt (recherche du meilleur mouvement et application de ce mouvement) s'effectue en $O(n^k)$.

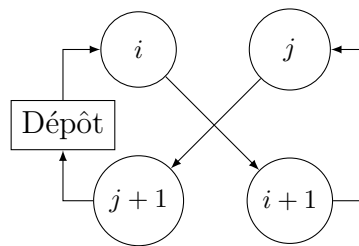


FIGURE 3.2 – Avant l’application de l’heuristique 2-opt

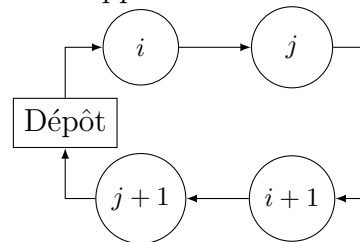


FIGURE 3.3 – Après l’application de l’heuristique 2-opt

Un mouvement Or-opt-1 considère chaque demande à son tour et essaie d’améliorer la solution en réinsérant cette demande à un autre emplacement. L’heuristique Or-opt étend cela en considérant des séquences de 1, 2 et 3 emplacements adjacents dans une solution.

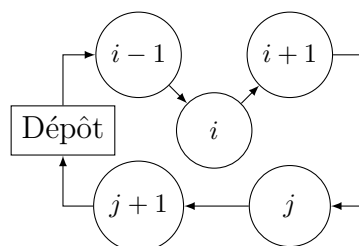


FIGURE 3.4 – Avant l’application de l’heuristique Or-opt

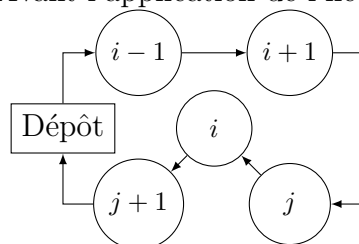


FIGURE 3.5 – Après l’application de l’heuristique Or-opt

3.3 Méta-heuristiques

Les méta-heuristiques sont des paradigmes d'optimisation. Elles ont le grand avantage d'être capable de s'adapter à différents types de problèmes combinatoires de plus grandes tailles ainsi que de très nombreuses applications qu'il était impossible de traiter auparavant. Toutes ces méta-heuristiques sont inspirées par des analogies : avec la physique, nous retrouvons le recuit simulé, avec la biologie nous retrouvons les algorithmes génétiques et avec l'éthologie, nous retrouvons les colonies de fourmis.

Leur objectif principal est de surmonter les limitations des heuristiques classiques, en particulier leur possibilité de tomber dans l'optimum local et leur manque de robustesse. Lors de l'exploration de l'espace des solutions elles combinent deux techniques : l'intensification qui permet de trouver l'optimum local de la solution courante, et diversification permet de sortir de la région de ce minimum local dans le but d'obtenir probablement un optimum global qui se trouve ailleurs. Parmi ces méthodes nous citons :

3.3.1 Algorithmes génétiques

Des algorithmes génétiques ont été développés par J. Holland dans les années 1970 pour comprendre les processus d'adaptation des systèmes naturels [7]. Cette méthode bio-inspirée a été formalisée ensuite par Goldberg [1989] pour être appliquée à la résolution de problèmes d'optimisation.

L'idée de base des algorithmes évolutifs au sein d'une population de solutions échangeant certaines informations pour produire de nouvelles solutions mieux adaptées. En partant avec une population initiale, souvent générée aléatoirement, on applique ces trois stratégies séquentielles :

1. **Stratégie de sélection** : La stratégie de sélection aborde la question suivante : "Quels parents de la prochaine génération sont choisis avec un parti pris pour une meilleure forme physique?".
Le principe de base des méthodes de sélection est le suivant : "Plus un individu est performant, plus il a de chance d'être parent". Une telle pression de sélection poussera la population à trouver de meilleures solutions.
2. **Stratégie de reproduction** : La stratégie de reproduction consiste à concevoir un ou plusieurs opérateurs de mutation et de croisement appropriés pour générer de nouveaux individus avec de nouvelles potentialités dans la génération future.
3. **Stratégie de remplacement** : les nouveaux descendants sont en concurrence avec des personnes âgées pour leur place dans la génération suivante (survie pour le plus apte).

Dès que la condition d'arrêt est vérifiée, on aura une solution améliorée de la population initiale. L'algorithme se présente comme suit [17] :

Algorithme 1 Algorithme génétique

```

1: popul, nouvpop, qualité : tableaux de taille  $N$  ;
2: popul := initialiser une population ;
3: répéter
4:   QUALITÉ (Évaluer la population popul) ;
5:   répéter
6:     (phase de reproduction génétique) ;
7:     Sélection ;
8:     Croisement ;
9:     Mutation ;
10:  jusqu'à popul remplie ;
11:  popul := Sélectionner nouvelle population (popul, nouvpopul) ;
12:   $f(x) := \min_{1 \leq i \leq N} [f(x_i)]$ ;
13:  si  $f(x) < f_{min}$  alors
14:     $f_{min} := f(x)$ ,  $x_{min} := x$ ;
15:  fin;
16: jusqu'à condition d'arrêt ;

```

Ando et Taniguchi (2006) utilisent un algorithme génétique pour résoudre le CVRP avec des fenêtres de temps doux et des temps de parcours stochastiques. Cependant, une description détaillée de l'algorithme n'est pas présentée. La qualité des solutions est évaluée par simulation. La solution la mieux trouvée est comparée à l'opération habituelle pendant cinq jours. La solution trouvée par l'algorithme fonctionne mieux. La moyenne du coût total a été réduite d'environ 4% et son écart-type d'environ 75% [19].

3.3.2 Récuit simulé

La méthode de recuit simulé a été découverte en 1983 par S. Kirkpatrick et al. Ils se sont inspirés du phénomène de la thermodynamique appliqué en physique statique "métallurgie".

En effet, la température T représente un paramètre de commande permettant au physicien de modifier l'état d'un matériau. De ce fait, afin d'obtenir une structure solide, organisée et cristalline, un réchauffement local doit être effectué suivi d'un refroidissement progressif afin d'atteindre un état parfaitement stable. En effet, si l'on refroidit brusquement un matériau en fusion, celui-ci présentera de nombreux défauts, c'est l'équivalent d'un optimum local pour un problème d'optimisation combinatoire. C'est pour cette raison que la descente de température se fait progressivement, les atomes se réarrangent, les défauts disparaissent et le métal a alors une structure ordonnée, c'est l'équivalent d'un optimum

global.

Par analogie à ce processus, on applique la méthode du recuit simulé à des problèmes d'OC NP-difficiles pour échapper à un optimum local, et ce en acceptant, avec une certaine probabilité $P = \exp^{-\left(\frac{\Delta f}{T}\right)}$, la dégradation de la fonction objectif f , afin de trouver une solution plus efficace que celle en cours.

La température diminue lentement au fur et à mesure du déroulement de l'algorithme de recuit simulé qui est présenté comme suit :

Algorithme 2 Recuit simulé

```

1: Entrées Insérer
   % Programme de refroidissement
2:  $s \leftarrow s_0$  ; %générer la solution initiale
3:  $T \leftarrow T_{max}$  ; %générer la température initiale
4: répéter
5:   répéter
6:     %a une température fixée
7:     Choisir aléatoirement  $s'$  voisin de  $s$  ;
8:      $\nabla E \leftarrow f(s') - f(s)$  ;
9:     si  $\nabla E < 0$  alors
10:       $s \leftarrow s'$  ; %solution améliorée
11:     sinon
12:       $s \leftarrow s'$  ;
13:      avec une probabilité de  $\exp \frac{-\nabla E}{T}$  ;
14:     finsi ;
15:   jusqu'à satisfaction de l'équilibre ;
16:   %fixer le nombre d'itération à exécuter pour chaque température
17:    $T \leftarrow g(T)$  ; %réduire la température
18: jusqu'à satisfaction du critère d'arrêt ; %  $T \leftarrow T_{max}$ 

```

3.3.3 Méthode de recherche tabou

Cette méta-heuristique est initiée, par Glover et Hansen en 1986. Elle s'appuie sur la notion d'histoire dans la politique d'exploration des solutions, inspirée de l'amélioration itérative, elle est basée sur le principe suivant : "garder des traces du passé pour mieux s'orienter dans le futur".

La méthode effectue une exploration de l'espace de solutions en passant d'une solution s identifiée dans l'itération S à la meilleure solution s' dans un sous-ensemble du voisinage $N(s)$ de s . Puisque s' ne s'améliore pas nécessairement sur s , un mécanisme de tabou est mis en place pour empêcher le processus de faire un cycle sur une suite de solutions. Une façon incomparable d'empêcher les cycles est d'interdire le processus de revenir aux solutions précédemment rencontrées.

Une description formelle du processus est la suivante :

Algorithme 3 Algorithme de recherche tabou

```

1: Initialisation
2: Trouver une solution initial  $x_0$  ;
3:  $x^* \leftarrow x_0, c^* \leftarrow f(x_0)$  ;
   %  $x^*$  est la meilleure solution rencontrée.
   %  $c^*$  son coût, et  $f$  la fonction de gain.
4:  $k \leftarrow 0, Listetabou = \emptyset$  ;
5: répéter
6: Choisir parmi le voisinage de  $x_k, V(x_k)$ , le mouvement qui minimise  $f$  et qui n'appartient pas à la Liste Tabou meilleur  $x_k$  ;
7:  $x_{k+1} \leftarrow$  meilleur  $x_k$  ;
8:
9: si ( $c(x_{k+1}) < c^*$ ) alors
10:    $x^* \leftarrow x_{k+1}$  ;
11:    $c^* \leftarrow (c(x_{k+1}))$  ;
12: finsi ;
13: Mise à jour de la Liste Tabou ;
14: jusqu'à Critère d'arrêt est vérifié ;

```

3.3.4 Optimisation par Colonies de fourmis



C'est une nouvelle méthode de résolution des problèmes combinatoires, et qui va être utilisée pour la résolution de notre problème. Les colonies de fourmis sont basées sur le comportement réel des fourmis à la recherche de nourriture. Ce comportement, décrit dans (Deneubourg et al., 1990), leur permet de trouver les plus courts chemins entre les sources de nourriture et leur nid(les noeuds). Ceci est possible grâce à la trace chimique (appelée phéromone) qui est une substance que chacune des fourmis dépose sur le sol. Lorsqu'une fourmi doit choisir entre deux directions, la bonne route est avec une plus grande probabilité celle comportant une plus forte concentration de phéromones. Les fourmis qui empruntent les sentiers les plus courts arrivent rapidement à se procurer de la nourriture et rentrent au nid en déposant de la phéromone sur leur chemin de retour. De plus, ces phéromones s'évaporent au cours du temps. Cela efface progressivement les traces les moins fréquentées. Notons que lorsque la trace de phéromone est trop faible, l'exploration des fourmis devient complètement aléatoire[24].

Fonctionnement de l'algorithme de base Ant system(AS) :

Le problème de base qui est le TSP est généralement défini comme un graphe complètement connecté $G = (V, E)$ où : V : est l'ensemble des noeuds (arrêts), E : le trajet entre ces

arrêts.

La procédure se déroule de la manière suivante : l'ensemble de la colonie de fourmis est placée au noeud de départ (dépôt central, qui est dans notre cas l'Université). Elle se déplace d'un arrêt à un autre, afin de former des tournées qui ne dépasse pas le temps maximal (T_{max}^k) tout en satisfaisant les exigences des étudiants.

Les variables que nous devons définir pour comprendre la suite sont les suivantes :

- $b_i(t)$ (ou $i \in V$) le nombre de fourmis dans l'arrêt i à l'instant t ;
- $m = \sum_{i=0}^n b_i$ le nombre total des fourmis, invariant dans le temps ;
- $\tau_{ij}(t)$, la quantité de phéromones τ_{ij} , à l'instant t , déposée sur l'arête reliant les deux arrêts : appelée l'intensité de la piste. Ce paramètre définit l'attractivité d'une partie du trajet global et change à chaque passage d'une fourmi ;
- $n = |V|$, le nombre des arrêts ;
- $\eta_{ij} = \frac{1}{d_{ij}}$, la visibilité de l'arrêt j quand on est placé sur l'arrêt i , invariante dans le temps. Elle est utilisée pour diriger le choix des fourmis vers des arrêts proches, et éviter les arrêts trop lointaines ;
- S^k , la liste des arrêts qui ne sont pas visités par la fourmi k placée sur l'arrêt i . Cela permet de garantir qu'aucune fourmi ne visitera deux fois un même arrêt au cours de sa recherche. La mémoire de chaque fourmi est vidée lorsqu'elles ont terminé leur cycle.

Choix des déplacements

Le passage de chaque fourmi K d'un arrêt i à un arrêt j à l'instant t , en fonction de la visibilité η_{ij} de cet arrêt et de la quantité de phéromones $\tau_{ij}(t)$ déposée sur l'arc reliant ces deux arrêts. Cela se fait aléatoirement avec une probabilité donnée par l'équation (3.6) suivante :

$$P_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij}(t))^\alpha \cdot (\eta_{ij})^\beta}{\sum_{L \in V \setminus S^k} (\tau_{iL}(t))^\alpha \cdot (\eta_{iL})^\beta} & \text{si } j \in S^k; \\ 0 & \text{sinon.} \end{cases} \quad (3.6)$$

où : α et β sont deux paramètres qui contrôlent l'importance relative entre phéromones et visibilité. Avec $\alpha = 0$, seule la visibilité de la ville est prise en compte, l'arrêt le plus proche est donc choisi à chaque pas. Au contraire, avec $\beta = 0$, seule les pistes de phéromone comptent.



Mise à jour des phéromones

Les variables des phéromones sont mises à jour selon la formule suivante :

$$\tau_{ij}^{t+1} = \rho\tau_{ij}^t + \sum_{k=1}^m \Delta\tau_{ij}^k(t); \quad (3.7)$$

où : $\rho \in [0, 1[$ est le coefficient qui définit la vitesse d'évaporation des traces de phéromone présentes sur les arêtes.

$\Delta\tau_{ij}^k(t)$: représente le cumul de phéromone. Cette valeur est donnée par :

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{si } (i, j) \in V \setminus S^k; \\ 0 & \text{sinon.} \end{cases} \quad (3.8)$$

où Q est un paramètre constant.

Cet organigramme représente le fonctionnement de l'Algorithme ACO :



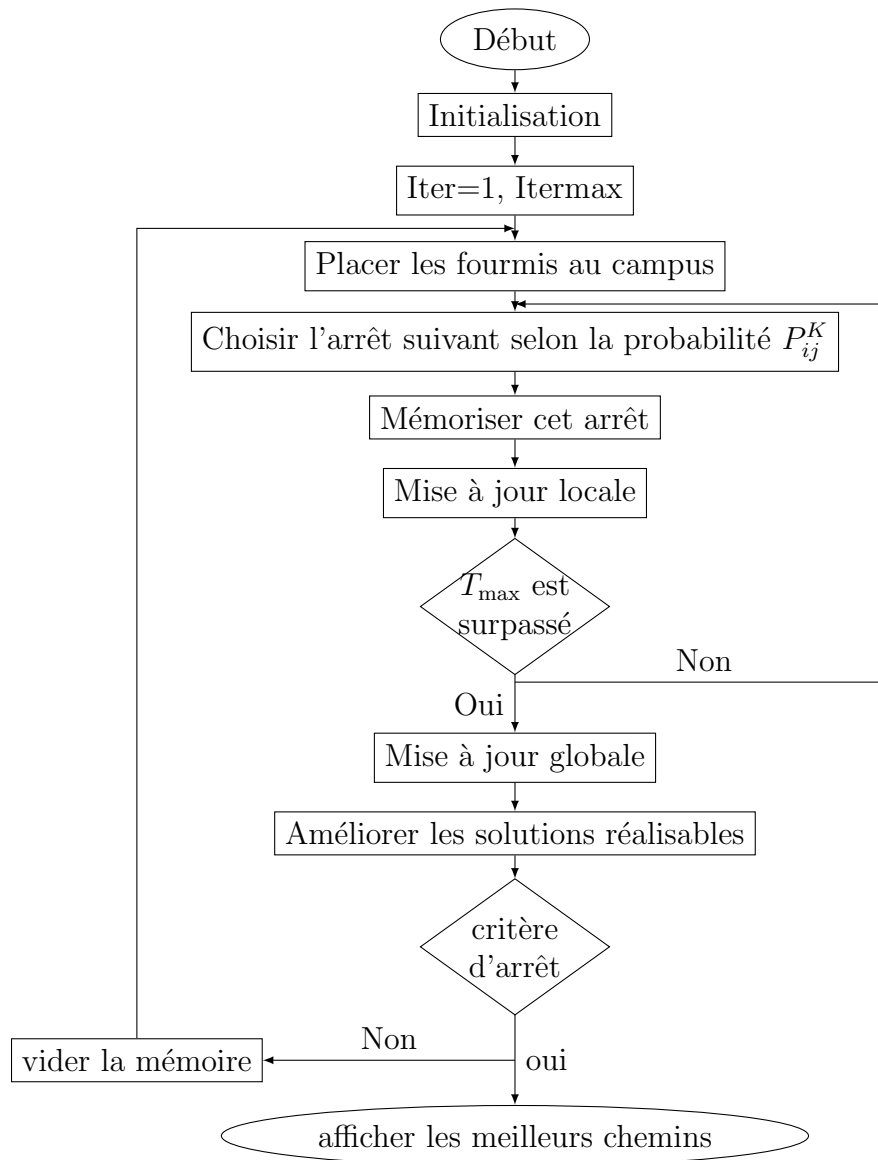


FIGURE 3.6 – Organigramme du fonctionnement de l'Algorithme ACO

Algorithme de base :

Soit :

t_{ij} : le temps de parcours de l'arc (i, j) ;

LS^k : liste de solution de la fourmi k .



Algorithme 4 Algorithme de colonies de fourmis

```

1: Initialisation
2:  $\tau_0 = c$ , non nul ;
3: Placer les fourmis dans le dépôt(Université) qui a l'indice 0 ;
4: iter=1 ;
5: pour  $i = 0 : n - 1$  faire
6:   pour  $j = 1 : n$  faire
7:     si  $j \in S^k$  alors
8:       pour  $k = 1 : m$  faire
9:         Sélectionner le prochain arrêt à visiter avec la probabilité  $P_{ij}^k = \frac{1}{|S^k|}$  ;
10:        si  $t_{0j} \leq T_{max}^k$  alors
11:          Insérer l'arrêt  $j$  dans  $LS^k$  ;
12:        finsi ;
13:      fin pour ;
14:    finsi ;
15:  fin pour ;
16: fin pour ;
17: pour  $iter = 2 : itermax$  faire
18:   pour  $i = 0 : n - 1$  faire
19:    pour  $j = 1 : n$  faire
20:      si  $j \in S^k$  alors
21:        pour  $k = 1 : m$  faire
22:          si  $t_{0j} \leq T_{max}^k$  alors
23:            Sélectionner le prochain arrêt à visiter avec la probabilité citée dans la
                formule (3.6) ;
24:            Insérer l'arrêt  $j$  dans  $LS^k$  ;
25:            Mise à jour des traces des phéromones selon la formule (3.7) ;
26:          finsi ;
27:        fin pour ;
28:      finsi ;
29:    fin pour ;
30:  fin pour ;
31: fin pour ;
32: Afficher les tournées résolues ;

```

Conclusion

Au cours de ce chapitre, nous avons décrit les principales méthodes de résolution du VRP et plus exactement l'algorithme de colonie de fourmis qui est une méta-heuristique utilisée avec succès pour résoudre plusieurs problèmes d'optimisation combinatoire. Ainsi, on l'a utilisé pour la résolution de notre problème de tournées des bus universitaires pour avoir une bonne affectation pour les bus, en introduisant l'algorithme d'optimisation par colonies de fourmis, son application pour un problème de SBRP fait l'objet du prochain chapitre.

4

Application

Contents

Introduction	41
4.1 Présentation de la direction des œuvres universitaires	42
4.2 Position du problème	42
4.3 Collecte de données	43
4.4 Implémentation et résultats	47
4.5 Comparaison des résultats avec l'organisation actuelle du transport universitaire	57
Conclusion	58

Introduction

Dans ce chapitre nous allons nous intéresser à la présentation et à l'implémentation de notre application pour le cas des bus universitaires, sous l'environnement MATLAB, qui est un langage de programmation très riche, utilisé largement pour le développement des applications de type très divers.

Ce chapitre est clôturé par une analyse des résultats obtenus. L'analyse est basée sur la comparaison entre les résultats obtenus avec l'algorithme de colonie de fourmis.

4.1 Présentation de la direction des œuvres universitaires

Les oeuvres universitaires de la wilaya de Béjaia sont gérées par deux directions, l'une sise à la commune de Béjaia et l'autre au niveau d'El Kseur, ayant pour mission l'amélioration des conditions de la vie estudiantine. Elle est dotée de différents départements et chacun est chargé des différentes prestations au service des étudiants. On distingue :

- Le département des bourses ;
- Le département des ressources humaines ;
- Le département des finances et des marchés publics ;
- Le département du contrôle et de la coordination.

Le département du contrôle et coordination, qui est chargé d'élaborer les plans de transport universitaire. Il comprend les services suivants :

- Le service de l'hébergement ;
- Le service de la restauration ;
- Le service du transport ;
- Le service des activités culturelles et sportives.

L'hébergement des étudiants résidant à plus de 50 km pour les garçons et plus de 30 Km pour les filles, de leurs lieux d'étude se fait au niveau de l'une des résidences universitaires suivantes :

- Résidence universitaire Targa Ouzemour ;
- Résidence universitaire 1000 lits ;
- Résidence universitaire 17 octobre 1961 ;
- Résidence universitaire Iheddadene ;
- Résidence universitaire Iryahene ;
- Résidence universitaire Berchiche 1,2,3,4 ;
- Résidence universitaire Amizour.

Le service du transport est chargé de planifier le transport des étudiants résidents dans les résidences universitaires ainsi que ceux habitant dans un rayon de 50 km des pôles universitaires.

4.2 Position du problème

La planification de tournées de bus universitaires peut être considérée de deux points de vue : celui du service de transport de la direction des œuvres universitaires de Béjaia et celui des étudiants. Le service de transport vise à générer une planification ne lui coûte pas trop cher, tout en assurant une qualité de service acceptable. Les étudiants quant à eux ne s'intéressent qu'à la qualité de service et l'amélioration de leur mobilité estudiantine.

Il est plus judicieux de considérer le problème du point de vue des étudiants.

Les principaux indicateurs contribuant à satisfaire les étudiants sont :

- Ne pas avoir à attendre trop longtemps dans l'arrêt des COUS ou, entre leur arrivé au campus et l'heure de début des cours. Ce qui revient à minimiser le temps d'attente à l'arrêt ou bien au campus ;
- Augmenter la fréquence de passage des COUS, et cela par l'augmentation de nombre du COUS à disposition ;
- Ne pas avoir un temps de trajet trop long(temps passé à l'interieur du COUS).

Après avoir fait une enquête au près des étudiants, on a constaté que la plupart d'entre eux ont choisit le temps de trajet comme le point le plus urgent à améliorer. Du fait, dans notre étude, nous nous sommes basés sur la minimisation du temps total de tournées des bus universitaires.

Actuellement le transport universitaire s'effectue de la manière suivante : Chaque résidence dispose d'un certain nombre de bus qui ont une capacité homogène égale à 100. Ces bus font des navettes entre les résidences et les pôles universitaires, sans pour autant effectuer des arrêts intermédiaires. Pour le transport des étudiants non résidents, des bus similaires effectue des trajets prédéterminés à des horaires fixes.

Si on prend par exemple le cas de la résidence universitaire 1000 lits, 11 bus effectuent chacun 7 rotations vers le campus Abderehmane MIRA, il mettent 15 minutes en moyenne pour une rotations, commençant à partir de 7h00mn jusqu'au 18h.

L'approche que nous avons utilisé dans ce travail, nous permet d'avoir une planification optimal qui réduit le temps de trajet avec moins de bus(un bus visite plus d'arrêt dans un temps minimum).

4.3 Collecte de données

À partir des deux directions des oeuvres universitaire de Béjaia, nous avons obtenu les principaux arrêts du réseau de transport universitaire et les points de départ pour chaque trajet situé en 74 lieux géographiques dont les 12 cités universitaires, deux arrêts campus, et 64 arrêts intermédiaires comme indiqué ci-dessous :

Pole Targa Ouzemour
Pole Aboudaou
RU Pépinière
RU 1000 Lits
RU 17 Octobre
RU Targa-Ouzemour
RU Iheddaden
RU Berchiche 1
RU Berchiche 2
RU Berchiche 3
RU Berchiche 4
RU Amizour 1
RU Amizour 2
RU Ireyahen

TABLE 4.1 – Arrêts universitaires

Ighil Ouazoug	Melalla
300 logts	Ibachiren
mosqué Iheddaden	Boukhelifa
BNA	Sidi Ayad
la Zone	Fenaïa
El-Kods	Timezrit
Route de Sétif	Semaoun
Oued Ghir	El Kseur
4 Chemins	Adekar
Tobal	Toudja
la Wilaya	Amizour Ville
Dawadji	Feraoun
Boukhiamama	Barbacha
1000 logts	Taslent
Mangin	Sebt aqudim
Sidi Ouali	La poste
Taassaste	Merdj Ouamane
Sidi ahmed	Champion
Polyvalent	Domaine El-Aifa
J3 Sidi-Ahmed	Bouseltane
Derguina	Tala Hamza
Souq El-Tnine	Abrouak
Melbou	Djamaa Oussaka
Aokas	Tizi
Baccaro	Ireyahen
Tichy	Agarage
Lota	Lamber
Ait Anane Ibelhadjen	Ibourassen
Tizi-N'berber	Larbaa
Centre	Ibelhadjen
Parc Communal	Lota
Para Medicale	Ighil El Bordj
Oussama	Antiq N'tafath

TABLE 4.2 – Arrêts intermédiaires

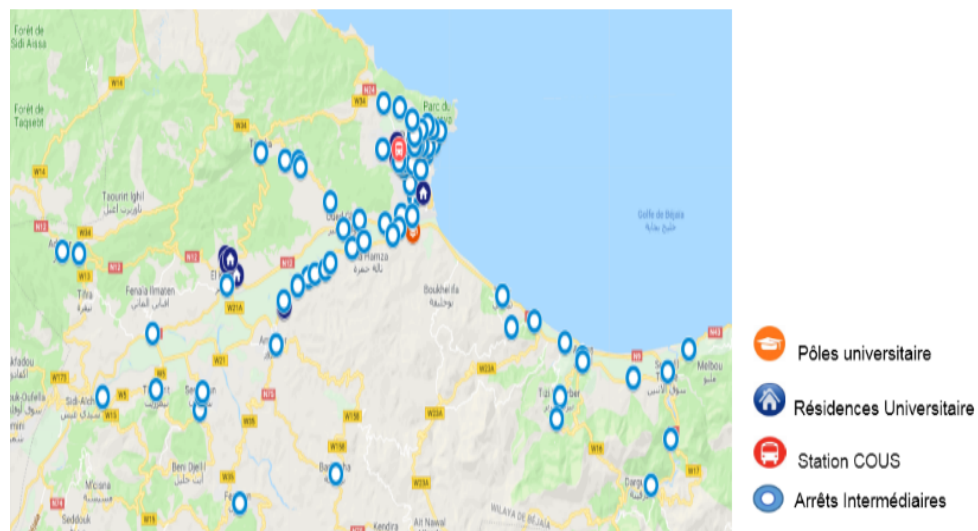


FIGURE 4.1 – Présentation géographique des arrêts des bus universitaires

Les arrêts qu'on a pris en considération dans notre travail, tel qu'illustrés sur la Figure(4.2), où le nombre d'arrêts est 20, dont celui du Pôle universitaire Targa Ouzemour, 6 résidences universitaires et 13 arrêts externes.

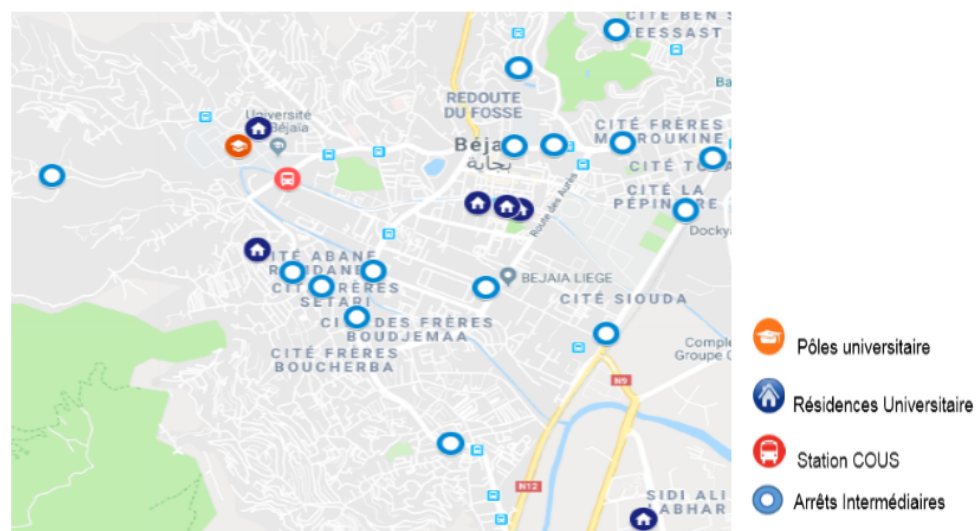


FIGURE 4.2 – Présentation géographique des arrêts des bus universitaires : pris dans notre étude

On associe à chaque paire d'arrêt un coût qui est typiquement le temps de trajet pour parcourir la distance qui les sépare.

Nous allons construire des chemins satisfaisant la contrainte :

$$\sum_{i=0}^n t_i^k \sum_{j=0}^n x_{ij}^k + \sum_{i=0}^n \sum_{j=0}^n t_{ij}^k x_{ij}^k \leq T_{max}^k \quad \forall k = 1, \dots, m. \quad (4.1)$$

Le procédé de la génération des chemins est le suivant : pour l'arrêt campus Targa Ouzemour(UTO) qui est le dépôt centrale v_0 on construit plusieurs chemins satisfaisant la contrainte de temps total (4.1). L'insertion d'un arrêt dans un chemin (tourné) est basée sur le principe d'augmentation minimal du temps de trajet total du chemin. Le sens de parcours du chemin étant jusqu'à présent indifférent dans le cadre de notre étude.

Nous avons calculé le temps de trajet qu'un bus parcourt entre le lieu de son départ à savoir le campus Targa Ouzemour(UTO) et le lieu de son arrivée. Nous avons numéroté les arrêts dans le Tableau (4.3).

1	Pole targa	11	300 logts
2	RU targa	12	BNA
3	RU 17 octobre	13	La zone
4	RU 1000 lits	14	4 chemins
5	RU pépinière	15	Polyvalent
6	Boukhama	16	Dawadji
7	RU iheddaden	17	La wilaya
8	RU ireyahen	18	Tobal
9	1000 logts	19	Ighil ouazoug
10	Mosquée iheddaden	20	Sidi ahmed

TABLE 4.3 – Arrêts du réseau

4.4 Implémentation et résultats

Après l'exécution de l'algorithme ACO sous le langage de programmation MATLAB, nous avons obtenu les résultats que nous avons reportés dans les tableaux suivants, où le nombre d'arrêts étudié dans notre cas est : 20 arrêts et le nombre de fourmis sont 200 et le nombre d'itérations est égale à 200. La valeur de chaque paramètre est : $\alpha = 1$; $\beta = 1$; $\rho = 0,5$; $\tau_0 = 2$. Nous avons supposé que la tournée ne doit pas dépasser le temps maximal $T_{max}^k = 1h$.

Nous avons traité les tournées dans les deux cas de circulation :

- cas fluide ;
- cas congestionné.

4.4.1 Cas fluide

Après avoir calculé les distances inter-arrêts, on a estimé les temps entre chaque paire d'arrêts supposant que le véhicule parcourt une vitesse de 30km/heure. Les tournées optimales (sans prendre en considération le retour puisque c'est le même que l'aller) et leurs temps de parcours sont comme suit :

Tournées	Temps de parcours(h)
1 → 8 → 6	0.4340
1 → 2 → 12 → 3 → 4 → 5 → 16 → 18 → 8	0.3633
1 → 3 → 4 → 12 → 18 → 8	0.3652
1 → 11 → 10 → 9 → 7 → 18 → 8	0.4125
1 → 12 → 13 → 14 → 17 → 8	0.3699
1 → 9 → 15 → 4 → 3 → 10 → 11 → 8	0.3701
1 → 16 → 9 → 8	0.3824
1 → 17 → 3 → 4 → 5 → 12 → 8	0.3721
1 → 19 → 16 → 8	0.4069
1 → 6 → 8	0.3830
1 → 20 → 15 → 19 → 8	0.3780
1 → 13 → 15 → 12 → 8	0.3830
1 → 10 → 9 → 6 → 8	0.3817
1 → 4 → 5 → 13 → 14 → 8	0.3738
1 → 15 → 9 → 10 → 12 → 3 → 8	0.3721
1 → 7 → 9 → 19 → 4 → 3 → 8	0.3767
1 → 18 → 15 → 8	0.3765
1 → 5 → 4 → 16 → 15 → 8	0.3719
1 → 14 → 19 → 8	0.3693

TABLE 4.4 – Tournées optimales dans le cas fluide

4.4.2 Cas congestionné

Afin de calculer les temps inter-arrêts on a utilisé la fonction BPR[13], définie par :

$$t(e) = t_f(e) \times \left(1 + \gamma \left(\frac{n(e)}{K(e)} \right)^\delta \right). \quad (4.2)$$

où chaque arc est caractérisé par son temps de parcours à l'état fluide $t_f(e)$ et sa capacité $K(e)$, qui représente le nombre de véhicules maximale que cette route peut supporter $n(e)$. γ et δ sont des paramètres relatifs à l'état de la route, en générale on prend $\gamma = 0.15$ et $\delta = 4$. Ce qui été pris dans les six premiers tableaux :

Tournées	Temps de parcours(h)
1 → 14 → 13 → 12 → 7 → 9 → 6	0.3681
1 → 7 → 9 → 3 → 4 → 5 → 11 → 10 → 2	0.3359
1 → 11 → 12 → 13 → 17 → 18 → 5 → 3 → 15	0.3381
1 → 15 → 11 → 7 → 9 → 10 → 4 → 5 → 16 → 12	0.4296
1 → 15 → 11 → 7 → 9 → 10 → 4 → 5 → 16 → 12	0.3672
1 → 2 → 6 → 8	0.3797
1 → 3 → 4 → 5 → 13 → 9 → 12 → 11 → 10 → 7	0.3612
1 → 10 → 11 → 19 → 8	0.3147
1 → 18 → 4 → 5 → 3 → 13 → 12 → 11 → 10 → 9 → 7	0.3726
1 → 19 → 8	0.2998
1 → 8 → 17 → 5	0.4247
1 → 20 → 16 → 10 → 11 → 12 → 13	0.2972
1 → 6 → 8	0.4270
1 → 4 → 5 → 17 → 15 → 8	0.4037
1 → 5 → 4 → 3 → 13 → 11 → 12 → 6	0.3650
1 → 17 → 3 → 4 → 5 → 7 → 9 → 10 → 12	0.4126
1 → 16 → 20 → 18 → 8	0.3764
1 → 13 → 12 → 11 → 10 → 9 → 3 → 5 → 4	0.4054
1 → 12 → 11 → 4 → 5 → 3 → 13 → 14 → 10 → 9 → 7	0.4103

TABLE 4.5 – Tournées optimales dans le cas 1 : $T_{max}^k = 1h$, $n/k = 0.9$

Tournées	Temps de parcours (h)
1 → 2 → 7 → 9 → 13 → 18 → 17 → 5 → 4 → 12 → 10 → 3 → 15 → 11 → 14 → 16 → 20	0.8137
1 → 16 → 17 → 3 → 15 → 11 → 9 → 10 → 7 → 6 → 12 → 13 → 4 → 5 → 14 → 19	0.8472
1 → 14 → 11 → 10 → 9 → 7 → 16 → 17 → 3 → 4 → 5 → 15 → 13 → 12 → 18 → 20	0.8664
1 → 10 → 5 → 16 → 18 → 17 → 4 → 3 → 9 → 7 → 12 → 11 → 14 → 19 → 8	0.8294
1 → 4 → 5 → 10 → 9 → 12 → 16 → 11 → 3 → 18 → 17 → 13 → 8	0.8521
1 → 7 → 4 → 3 → 13 → 16 → 15 → 10 → 11 → 12 → 14 → 18 → 5 → 17 → 20	0.8196
1 → 20 → 18 → 4 → 5 → 3 → 19 → 11 → 13 → 12 → 17 → 7 → 8	0.8350
1 → 15 → 13 → 19 → 7 → 9 → 10 → 11 → 12 → 4 → 3 → 18 → 5 → 2 → 6	0.8477
1 → 5 → 4 → 3 → 12 → 10 → 18 → 11 → 2 → 9 → 8	0.8592
1 → 12 → 7 → 9 → 15 → 5 → 10 → 18 → 3 → 4 → 13 → 14 → 8	0.8465
1 → 3 → 18 → 15 → 5 → 4 → 9 → 10 → 12 → 11 → 14 → 17 → 13 → 16 → 20	0.8410
1 → 9 → 19 → 17 → 18 → 3 → 12 → 5 → 4 → 11 → 13 → 16 → 8	0.8317
1 → 17 → 5 → 3 → 4 → 15 → 19 → 10 → 7 → 9 → 13 → 20 → 8	0.8595
1 → 8 → 19 → 17 → 15 → 4 → 5 → 10 → 12 → 11 → 13 → 6	0.9188
1 → 13 → 12 → 3 → 9 → 7 → 10 → 16 → 18 → 5 → 4 → 15 → 20 → 17 → 8	0.8703
1 → 6 → 9 → 7 → 15 → 14 → 13 → 18 → 12 → 8	0.8563
1 → 11 → 4 → 5 → 18 → 17 → 16 → 10 → 12 → 15 → 3 → 13 → 8	0.8371
1 → 18 → 4 → 16 → 3 → 17 → 5 → 13 → 11 → 12 → 19 → 8	0.8340
1 → 19 → 7 → 9 → 12 → 10 → 5 → 4 → 3 → 15 → 16 → 11 → 13 → 14 → 8	0.8560

TABLE 4.6 – Tournées optimales dans le cas 2 : $T_{max}^k = 1h$, $n/k = 0.95$

Tournées	Temps de parcours (h)
1 → 14 → 13 → 4 → 17 → 18 → 5 → 3 → 6 → 8	0.8502
1 → 4 → 5 → 20 → 18 → 3 → 15 → 11 → 7 → 9 → 14 → 8	0.8467
1 → 8 → 7 → 9 → 16 → 3 → 5 → 4 → 12 → 17 → 13 → 14	0.9059
1 → 15 → 4 → 5 → 12 → 13 → 10 → 9 → 3 → 18 → 7 → 16 → 17 → 20	0.8431
1 → 9 → 11 → 19 → 3 → 4 → 5 → 2 → 12 → 13 → 15 → 18 → 8	0.8395
1 → 3 → 4 → 5 → 16 → 17 → 18 → 12 → 2 → 13 → 14 → 15 → 20	0.8371
1 → 13 → 14 → 12 → 10 → 9 → 18 → 3 → 15 → 20 → 8	0.8712
1 → 20 → 4 → 17 → 18 → 9 → 10 → 12 → 13 → 7 → 19 → 11 → 8	0.8408
1 → 11 → 10 → 13 → 18 → 12 → 5 → 4 → 3 → 15 → 14 → 17 → 16 → 20	0.8285
1 → 7 → 5 → 4 → 10 → 9 → 13 → 12 → 3 → 2 → 15 → 11 → 6	0.8347
1 → 16 → 5 → 4 → 13 → 14 → 18 → 17 → 3 → 10 → 12 → 11 → 9 → 8	0.8430
1 → 2 → 18 → 17 → 16 → 3 → 4 → 5 → 14 → 13 → 12 → 7 → 9 → 11 → 10 → 15 → 20	0.8079
1 → 10 → 16 → 12 → 9 → 7 → 17 → 18 → 4 → 5 → 13 → 15 → 3 → 20	0.8478
1 → 5 → 4 → 3 → 10 → 15 → 13 → 18 → 12 → 14 → 16 → 17 → 20	0.8223
1 → 17 → 15 → 19 → 4 → 12 → 10 → 3 → 13 → 5 → 8	0.8520
1 → 6 → 2 → 20 → 16 → 15 → 17 → 8	0.8512
1 → 18 → 9 → 3 → 4 → 5 → 16 → 14 → 7 → 8	0.8550
1 → 19 → 17 → 5 → 4 → 11 → 3 → 18 → 20 → 8	0.8541
1 → 12 → 9 → 7 → 3 → 11 → 13 → 10 → 16 → 15 → 4 → 5 → 2 → 6	0.8497

TABLE 4.7 – Tournées optimales dans le cas 3 : $T_{max}^k = 1h$, $n/k = 1$

Tournées	Temps de parcours (h)
1 → 19 → 12 → 8	0.3887
1 → 20 → 15 → 18 → 8	0.3526
1 → 5 → 4 → 3 → 13 → 8	0.3616
1 → 8 → 20	0.4445
1 → 3 → 7 → 17 → 6	0.3670
1 → 14 → 17 → 8	0.4023
1 → 2 → 14 → 8	0.3614
1 → 7 → 13 → 5 → 4 → 8	0.3583
1 → 13 → 15 → 8	0.3592
1 → 9 → 2 → 8	0.3672
1 → 15 → 17 → 5 → 3 → 8	0.3662
1 → 11 → 14 → 8	0.3822
1 → 6 → 8	0.4207
1 → 10 → 9 → 16 → 8	0.3787
1 → 4 → 13 → 8	0.3606
1 → 16 → 20 → 8	0.3940
1 → 17 → 3 → 12 → 8	0.3689
1 → 18 → 15 → 8	0.4136
1 → 12 → 9 → 7 → 4 → 5 → 8	0.3701

TABLE 4.8 – Tournées optimales dans le cas 4 : $T_{max}^k = 0.5h$, $n/k = 0.9$

Tournées	Temps de parcours (h)
1 → 19 → 11 → 12 → 8	0.3789
1 → 20 → 14 → 8	0.3767
1 → 5 → 4 → 19 → 8	0.4032
1 → 6 → 8	0.4298
1 → 14 → 13 → 8	0.3819
1 → 9 → 7 → 10 → 18 → 8	0.3565
1 → 8 → 20	0.4541
1 → 3 → 14 → 8	0.3782
1 → 16 → 5 → 4 → 15 → 8	0.3660
1 → 12 → 11 → 19 → 16 → 7 → 10 → 9	0.3418
1 → 18 → 8 → 17	0.4670
1 → 2 → 14 → 8	0.3692
1 → 17 → 4 → 5 → 3 → 8	0.3578
1 → 11 → 9 → 3 → 8	0.3829
1 → 4 → 5 → 3 → 18 → 8	0.3503
1 → 10 → 11 → 15 → 8	0.3560
1 → 7 → 15 → 3 → 8	0.3543
1 → 15 → 16 → 17 → 8	0.3605
1 → 13 → 15 → 8	0.3670

TABLE 4.9 – Tournées optimales dans le cas 5 : $T_{max}^k = 0.5h$, $n/k = 1$

Tournées	Temps de parcours (h)
1→ 13→ 4→ 5→ 3→ 8	0.3609
1→ 20→ 8→ 13→ 12	0.4876
1→ 12→ 18→ 8	0.3430
1→ 8→ 20	0.4654
1→ 6→ 20→ 16	0.4696
1→ 4→ 11→ 8	0.3584
1→ 18→ 4→ 5→ 14→ 2	0.3761
1→ 10→ 11→ 4→ 5→ 3→ 8	0.3566
1→ 16→ 4→ 5→ 3→ 12→ 8	0.3565
1→ 17→ 4→ 8	0.3500
1→ 14→ 8→ 18	0.4914
1→ 15→ 13→ 8	0.3607
1→ 2→ 19→ 8	0.3516
1→ 11→ 13→ 12→ 8	0.3503
1→ 3→ 17→ 5→ 4→ 8	0.3881
1→ 9→ 18→ 8	0.3469
1→ 7→ 12→ 9→ 8	0.3519
1→ 5→ 4→ 15→ 8	0.3545
1→ 19→ 3→ 8	0.4150

TABLE 4.10 – Tournées optimales dans le cas 6 : $T_{max}^k = 0.5h$, $n/k = 1$

En changeant la valeur des paramètres γ et δ par $\gamma = 1$ et $\delta = 1$, on obtient les tournées optimales présentées dans les tableaux suivants :

Tournées	Temps de parcours(h)
1 → 11 → 15 → 4 → 5 → 3 → 10 → 12 → 8	0.7765
1 → 13 → 4 → 10 → 5 → 8	0.7796
1 → 9 → 7 → 5 → 4 → 3 → 13 → 18 → 8	0.7553
1 → 7 → 6 → 2 → 8	0.8379
1 → 4 → 19 → 11 → 8	0.7644
1 → 6 → 12 → 19 → 7 → 2	0.9151
1 → 10 → 2 → 4 → 8	0.7623
1 → 20 → 3 → 13 → 18 → 8	0.7434
1 → 2 → 20 → 8	0.7575
1 → 12 → 6 → 8	0.7739
1 → 17 → 12 → 5 → 4 → 8	0.7705
1 → 8 → 3 → 2	0.8784
1 → 15 → 4 → 5 → 11 → 13 → 12 → 8	0.7583
1 → 3 → 4 → 5 → 12 → 18 → 8	0.7385
1 → 18 → 4 → 5 → 16 → 8	0.7429
1 → 19 → 15 → 8	0.7680
1 → 5 → 3 → 12 → 4 → 16 → 17 → 8	0.7861
1 → 14 → 5 → 4 → 10 → 8	0.7996
1 → 16 → 3 → 18 → 4 → 8	0.7516

TABLE 4.11 – Tournées optimales dans le cas 7 : $T_{max}^k = 1h$, $n/K = 0.9$

Tournées	Temps de parcours(h)
1 → 17 → 6 → 20	0.8919
1 → 10 → 7 → 13 → 16 → 18 → 8	0.7491
1 → 9 → 4 → 5 → 18 → 16 → 8	0.7543
1 → 4 → 3 → 5 → 15 → 17 → 16 → 8	0.8170
1 → 5 → 7 → 9 → 8	0.7379
1 → 8 → 6 → 7	0.9880
1 → 15 → 20 → 8	0.7495
1 → 6 → 8	0.7469
1 → 14 → 15 → 8	0.7703
1 → 18 → 20 → 8	0.7829
1 → 12 → 7 → 5 → 4 → 17 → 8	0.7641
1 → 11 → 10 → 15 → 16 → 8	0.7977
1 → 3 → 5 → 18 → 13 → 8	0.7760
1 → 7 → 9 → 17 → 18 → 4 → 8	0.7554
1 → 16 → 7 → 8	0.7482
1 → 13 → 4 → 5 → 17 → 3 → 8	0.7764
1 → 2 → 7 → 4 → 3 → 5 → 15 → 8	0.7494
1 → 19 → 17 → 8	0.7623
1 → 20 → 18 → 4 → 5 → 19 → 8	0.7465

TABLE 4.12 – Tournées optimales dans le cas 8 : $T_{max}^k = 1h$, $n/K = 0.95$

Tournées	Temps de parcours(h)
1 → 8 → 6	0.8680
1 → 17 → 4 → 5 → 3 → 18 → 8	0.7369
1 → 2 → 20 → 8	0.7974
1 → 6 → 8	0.7660
1 → 3 → 9 → 7 → 10 → 12 → 11 → 8	0.7473
1 → 11 → 4 → 3 → 17 → 18 → 8	0.7636
1 → 4 → 5 → 16 → 3 → 10 → 12 → 8	0.8061
1 → 7 → 9 → 6 → 8	0.7410
1 → 15 → 13 → 12 → 4 → 5 → 3 → 8	0.7423
1 → 10 → 7 → 9 → 11 → 13 → 5 → 8	0.7458
1 → 19 → 3 → 4 → 12 → 8	0.7862
1 → 12 → 18 → 5 → 8	0.7259
1 → 20 → 7 → 8	0.7553
1 → 13 → 17 → 15 → 8	0.7460
1 → 14 → 15 → 8	0.7960
1 → 5 → 18 → 17 → 4 → 8	0.7601
1 → 9 → 4 → 18 → 5 → 8	0.7320
1 → 16 → 4 → 3 → 15 → 18 → 8	0.7378
1 → 18 → 14 → 7 → 3 → 4 → 12 → 10	0.7553

TABLE 4.13 – Tournées optimales dans le cas 9 : $T_{max}^k = 1h$, $n/K = 1$

4.4.3 Interprétation des résultats

On remarque qu'après avoir varier les paramètres, dans le cas fluide on a obtenu des tournées avec des temps raisonnable qui varient entre 20 et 25 minutes.

Or que dans le cas congestionné, quand on fixe le T_{max}^k à une heure on obtenient de grandes tournées avec un temps qui tend vers 55 minutes, mais quand on fixe le T_{max}^k à 0.5 heure on obtient généralement des petites tournées.

On remarque aussi que les tournées résultantes quand $n/K = 0.9$, et $\gamma = 0.15$ et $\delta = 4$ sont plus petites que celle obtenues quand $n/K = 0.9$, et $\gamma = 1$ et $\delta = 1$.

4.5 Comparaison des résultats avec l'organisation actuelle du transport universitaire

Dans nos résultats, on a obtenu des tournées qui passent par le maximum des d'arrêts intermédiaires dans un temps raisonnable. Ce qui permet de réduire le nombre de bus disponible au niveau des résidences.

Par contre, actuellement les bus universitaires passent par le minimum d'arrêts. Les étudiants

étant jamais satisfait par ces planifications, contraignant ainsi la DOUB(Direction des Œuvres Universitaire de Béjaia) de rajouter des bus, sans pour autant régler le problème. Nous présentons quelques tournées actuelles et nos tournées proposées :

Tournées actuelles	Temps de parcours(h)
[RU Pépinière, Pole Targa Ouzemour]	0.11
[El kods, Cité tobal, La wilaya, Daouadji, Pole Targa Ouzemour]	0.17
[Mangin, Sidi Ouali, Taassaste, Daouadji, Pole Targa Ouzemour]	0.25
[Taassaste, Sidi Ahmed, Daouadji, Polyvalent, Pole Targa Ouzemour]	0.33
Tournées proposées	Temps de parcours(h)
[Pole Targa Ouzemour,RU Targa, BNA,RU 17 Octobre RU 1000 lits,RU Pépinière,Daouadji,Tobal,RU Ireyahen]	0.3633
[Pole Targa Ouzemour, BNA,la zone,4 chemins,La Wilaya RU Ireyahen]	0.3699
[Pole Targa Ouzemour,RU Pépinière,RU 1000 lits Daouadji,Polyvalent,RU Ireyahen]	0.3719
[Pole Targa Ouzemour,Sidi Ahmed,Polyvalent Ighil Ouazoug,RU Ireyahen]	0.3780

TABLE 4.14 – Tournées actuelles

Conclusion

Vu l'immensité du réseau de transport universitaire de la ville de Béjaia et le manque de temps pour collecter les données suivantes :

- La distance entre deux arrêts ;
- Le temps de parcours entre chaque deux arrêts ;
- Le nombre d'étudiants à l'arrêt universitaire ;

On s'est limité à l'étude du réseau du transport universitaire du centre ville de Béjaia. Dans ce chapitre, en premier lieu, nous avons présenté la direction des œuvres universitaires. Par la suite, nous avons implémenter l'algorithme Colonie de fourmis sous le langage de programmation choisi (MATLAB), afin de trouver une affectation routière optimale des bus universitaires.

En second lieu, nous avons comparé et interprété les résultats obtenus dans les deux cas, fluide(hors heures de pointe) et congestionné(heures de pointe) avec ceux obtenus avec la politique de planification utilisée actuellement au niveau de la direction des œuvres universitaires. Enfin, d'après nos résultats obtenus, on a constaté que notre méta-heuristique nous a permis de réduire le temps total des tournées des bus universitaires dans la ville de béjaia en passant par le maximum d'arrêts, tout en satisfaisant les exigences des étudiants.

Conclusion générale

Le travail présenté dans ce mémoire traite le problème de tournées des bus universitaire de la wilaya de Béjaia, modélisé sous la forme d'un problème de tournée de véhicules. Afin de résoudre le modèle élaboré, une méta-heuristique, qui n'est autre que les colonies de fourmis a été utilisée.

Une des spécificités de ce travail se situe au niveau de la modélisation. En effet, nous avons placé les étudiants au centre du problème en nous focalisant sur la minimisation de leurs temps de parcours (minimisation du temps total des tournées tout en visitant le maximum d'arrêts), cette modélisation nous a permis des planifications qui permettent à un bus de transporter simultanément des étudiants devant se rendre dans le campus. Cette particularité a été d'un grand intérêt par rapport à la diminution de temps de parcours des étudiants.

Pour atteindre notre objectif nous avons utilisé une métaheuristique de colonie de fourmis. Cependant, les algorithmes de colonies de fourmis sont des algorithmes inspirés du comportement naturel des fourmis qui consiste à optimiser le trajet de la fourmilière à l'endroit de la nourriture, nous avons appliqué cette méthode pour optimiser le trajet des bus universitaires partant du campus vers les différents arrêts au cours de livraison des étudiants. L'application de cette approche nous facilite la planification et la régénération des routes en très peu de temps.

Dans notre cas, le réseau étudié est constitué d'un dépôt centrale qui est le campus universitaire (Targa Ouzemour), de plusieurs arrêts vaut exactement 19 arrêts (6 résidences universitaires et 13 arrêts externes), et d'une flotte composée d'un nombre limité de bus de capacité homogène.

Nous avons notamment fait une planification des deux cas : fluide et congestionné, on a trouvé de petites tournées avec des temps plus grands en cas de congestion, contrairement dans le cas fluide, on a trouvé des tournées qui passent par le maximum d'arrêts en un temps raisonnable. Ces résultats peuvent être appliquées dans la réalité afin d'améliorer le niveau de service des bus universitaires.

Bibliographie

- [1] AKLI, M. *Problème de tournées de véhicules avec contraintes et fenêtre de temps*. PhD thesis, UMMTO, 2013.
- [2] ARCHETTI, C., AND SPERANZA, M. G. The split delivery vehicle routing problem : a survey. In *The vehicle routing problem : Latest advances and new challenges*. Springer, 2008, pp. 103–122.
- [3] BANIEL, F. *Prise en compte d'objectifs de stabilité pour l'organisation de collectes de déchets*. PhD thesis, Institut National Polytechnique de Toulouse-INPT, 2009.
- [4] BELLMAN, R. Dynamic programming and stochastic control processes. *Information and control* 1, 3 (1958), 228–239.
- [5] BRÄYSY, O. *Local search and variable neighborhood search algorithms for the vehicle routing problem with time windows*. Vaasan yliopisto, 2001.
- [6] CORDEAU, J.-F., AND GROUPE D'ÉTUDES ET DE RECHERCHE EN ANALYSE DES DÉCISIONS (MONTRÉAL, Q. *The VRP with time windows*. Montréal : Groupe d'études et de recherche en analyse des décisions, 2000.
- [7] DRIDI, I. H. *Optimisation heuristique pour la résolution du m-PDPTW statique et dynamique*. PhD thesis, École Centrale de Lille, 2010.
- [8] ELDRANDALY, K. A., AND ABDALLAH, A. F. A novel gis-based decision-making framework for the school bus routing problem. *Geo-spatial Information Science* 15, 1 (2012), 51–59.
- [9] FU, Z., EGGLESE, R., AND LI, L. Y. A new tabu search heuristic for the open vehicle routing problem. *Journal of the operational Research Society* 56, 3 (2005), 267–274.
- [10] GENDREAU, M., GUERTIN, F., POTVIN, J.-Y., AND TAILLARD, E. Parallel tabu search for real-time vehicle routing and dispatching. *Transportation science* 33, 4 (1999), 381–390.
- [11] GURPREETSINGH, E., AND DHIR, V. Open vehicle routing problem by ant colony optimization. *International Journal of Advanced Computer Science and Applications* 5 (2014), 63–68.
- [12] HERRERA-GRANDA, I. D., LORENTE-LEYVA, L. L., PELUFFO-ORDÓÑEZ, D. H., VALENCIA-CHAPI, R. M., MONTERO-SANTOS, Y., CHICAIZA-VACA, J. L., AND CASTRO-OSPINA, A. E. Optimization of the university transportation by contraction hierarchies method and clustering algorithms. In *International Conference on Hybrid Artificial Intelligence Systems* (2018), Springer, pp. 95–107.

- [13] IDRES, L. *Application de la théorie des jeux multicritères à l'organisation d'un système de transport urbain*. PhD thesis, Université de Béjaia-Abderrahmane Mira.
- [14] ISMAIL, S. B., LEGRAS, F., AND COPPIN, G. *Synthèse du problème de routage de véhicules*. PhD thesis, Dépt. Logique des Usages, Sciences Sociales et de l'Information, 2011.
- [15] KAMMARTI, R. *Approches évolutionnistes pour la résolution du 1-PDPTW statique et dynamique*. PhD thesis, Ecole Centrale de Lille, 2006.
- [16] KIM, B.-I., KIM, S., AND PARK, J. A school bus scheduling problem. *European Journal of Operational Research* 218, 2 (2012), 577–585.
- [17] M. MAAFA, N. *Optimisation du schéma de distribution des carburants au niveau de NAFTAL*. PhD thesis, Université de bejaia, 2003.
- [18] MARCHAND, H. *Optimisation d'un problème de tournées de véhicules se posant dans l'industrie du carton*. PhD thesis, École des hautes études commerciales, 2006.
- [19] OYOLA, J., ARNTZEN, H., AND WOODRUFF, D. L. The stochastic vehicle routing problem, a literature review, part i : models. *EURO Journal on Transportation and Logistics* 7, 3 (2018), 193–221.
- [20] PARK, J., AND KIM, B.-I. The school bus routing problem : A review. *European Journal of operational research* 202, 2 (2010), 311–319.
- [21] SMAILI, Y. *Optimisation des flux dans un système de distribution-Application à une entreprise*. PhD thesis, Université de Batna 2-Mustafa Ben Boulaid, 2012.
- [22] SPADA, M., BIERLAIRE, M., AND LIEBLING, T. M. Decision-aiding methodology for the school bus routing and scheduling problem. *Transportation Science* 39, 4 (2005), 477–490.
- [23] TAILLARD, É. D. A heuristic column generation method for the heterogeneous fleet vrp. *RAIRO-Operations Research* 33, 1 (1999), 1–14.
- [24] TALBI, E.-G. *Metaheuristics : from design to implementation*, vol. 74. John Wiley & Sons, 2009.
- [25] YAO, B., CAO, Q., WANG, Z., HU, P., ZHANG, M., AND YU, B. A two-stage heuristic algorithm for the school bus routing problem with mixed load plan. *Transportation Letters* 8, 4 (2016), 205–219.

Résumé

Résumé : Développement de Méta-heuristique pour la Résolution du Problème de Tournées de Véhicules : Application au Transport Universitaire.

*L*e service du transport universitaire a pour objectif d'acheminer les étudiants de leurs lieux de résidences à leurs campus et vice versa. Afin d'apporter une solution à ce problème combinatoire complexe une métaheuristique permettant de construire une planification de tournées de bus universitaires de bonne qualité est présenté dans ce travail. Nous avons modélisé le problème de planification des tournées des bus universitaires. Ce problème est exprimé sous la forme d'un programme linéaire à variables binaires. Nous avons utilisé l'algorithme de colonie de fourmis pour l'obtention d'une affectation optimale des tournées des bus universitaires. Cette approche nous a permis de construire des planifications bien meilleure que celles utilisées par le service du transport de la direction des oeuvres universitaires de Béjaia.

Mots clés : Tournées de véhicules (VRP), SBRP(School Bus Routing Problem), colonie de fourmis.

Abstract : Development of Meta-heuristics for the Resolution of the Problem of Vehicle Tours : Application to University Transport.

*T*he purpose of the university transportation service is to transport students from their residency to campus and vice versa. In order to provide a solution to this complex combinatorial problem a metaheuristic allows to build a planning of university bus tours of good quality is presented in this work. We have modeled the problem of planning university bus tours. This problem is expressed as a linear program with binary variables. We used the ant colony algorithm to obtain optimal assignment of university bus tours. this approach allowed us to build much better plans than those used by the transport department of the management of Béjaia's academic works.

Keywords : Vehicle Tours (VRP), hybrid algorithm, SBRP(School Bus Routing Problem), colony of ants.
