

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique



Département Génie Industriel

Mémoire de projet de fin d'études Pour l'obtention du diplôme d'ingénieur d'état en Génie Industriel

Automatisation du service de livraison de Yassir Foods.

Oussama CHEKAI

Sous la direction de Mme. Wedjdane Nahili

MAB

Présenté et soutenu publiquement le (10/07/2021)

Composition du Jury :

Président	Dr. Okba Hamri	MCA	ENP
Promoteur	Mme. Wedjdane Nahili	MAB	ENP
Examineur	Dr. Hakim Fourar Laidi	MCB	ENP

ENP 2021

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique



Département Génie Industriel

Mémoire de projet de fin d'études Pour l'obtention du diplôme d'ingénieur d'état en Génie Industriel

Automatisation du service de livraison de Yassir Foods.

Oussama CHEKAI

Sous la direction de Mme. Wedjdane Nahili

MAB

Présenté et soutenu publiquement le (10/07/2021)

Composition du Jury :

Président	Dr. Okba Hamri	MCA	ENP
Promoteur	Mme. Wedjdane Nahili	MAB	ENP
Examineur	Dr. Hakim Fourar Laidi	MCB	ENP

ENP 2021

Dédicace

Je dédie ce travail à mes deux chers parents pour m' avoir soutenu durant toutes les étapes de ma vie.

A mon frère Rami qui m' a toujours aidé à prendre les bonnes décisions.

A l' équipe Collo pour les bons moments passé ensemble durant notre formation à l' ENP.

Oussama.

Remerciements

Ce travail est le fruit d'intervention et la contribution d'un grand nombre de personnes.

Je tiens à remercier Mme WEDJDANE NAHILI pour l'aide et l'orientation qu'elle a apporté lors de la réalisation de ce travail.

Je souhaite aussi adresser mes remerciements à Monsieur/Madame membres du jury, pour l'intérêt qui ont porté à ce travail et le temps d'examiner ce projet.

Je tiens à remercier mon promoteur en entreprise Mme THILELI ZEHRAOUI pour l'accueil et son support inconditionnel.

Mes vifs remerciements vont aussi à toute l'équipe de Yassir, particulièrement à l'équipe Sales et Data pour leur support continu.

ملخص

الهدف من هذا العمل هو اقتراح أتمتة عملية تسليم ياسير. باستخدام توزيع الموارد اللوجستية، وتطوير النماذج الرياضية ونهج القرار العلمي، سيتم تطوير تكنولوجيا المعلومات لأتمتة عملية التسليم يتم تحقيق باستخدام خوارزمية للاستمثال تم إجراء بحث لتحديد احتياجات المؤسسة، ووضع تصور لنموذج رياضي واقتراح طريقة لتنفيذ البرنامج الحاسوبي

الكلمات المفتاحية: اللوجستيات، بحوث العمليات، الأدلة العليا، الاستمثال

Abstarct

The objective of this work is to propose an automation of the delivery process of Yassir; using a scientific resolution approach, an information technology solution will be developed to automate the delivery process. The automation of the delivery process is achieved using an optimization algorithm.

A research approach was conducted to identify the needs of the enterprise, conceptualize a mathematical model and propose an approach to implement the software.

Keywords: Logistics, Operations Research, Metaheuristics, Optimization.

Résumé

L'objectif de ce travail est de concevoir un outil d'automatisation de l'affectation des livraisons de l'entreprise Yassir ; à l'aide d'une approche de résolution basée sur la recherche scientifique, un outil informatique d'aide à la décision sera développé pour permettre l'automatisation de la livraison. L'automatisation de la livraison est achevée à l'aide d'un algorithme d'optimisation.

Une démarche de recherche a été conduite et a permis d'identifier le besoin de l'entreprise, conceptualiser le modèle mathématique et proposer une approche pour la mise en place de l'outils suggéré.

Mots-clés : Logistique, Recherche opérationnelle, métaheuristique, Optimisation.

Table des matières

Liste des tableaux

Liste des figures

Liste des abréviations

Introduction générale :.....	11
1 Économie des plateformes et son évolution en Algérie	12
1.1 L'économie des plateformes :.....	13
1.2 Yassir : une plateforme Algérienne :	16
1.3 Activité et organisation de Yassir en Algérie :	18
1.4 Diagnostique du marché de l'entreprise :	23
1.5 Diagnostic stratégique des services de Yassir :	23
1.6 Analyse externe suivant les cinq forces de Porter :	24
1.7 Analyse interne par l'approche processus :.....	25
1.8 Résultat du diagnostic interne :.....	33
1.9 Description de la problématique :	35
2 Chapitre 2 : introduction au VRP et ses problèmes dérivés	36
2.1 Introduction :.....	37
2.2 Problèmes des tournées de véhicule :	37
2.3 Les variantes du VRP dans la littérature :.....	39
2.4 Problème de tournées de véhicule avec ramassage :.....	43
2.5 Résolutions des VRPs :.....	45
2.6 Méthodes de résolution exactes :	45
2.7 Branch and Bound :	46
2.8 Branch and Cut:	48
2.9 Branch and Cut and Price:	49

2.10	Conclusion :	49
2.11	Méthodes de résolution approximatives :	49
2.12	Approches Métaheuristiques :	50
2.13	Recherche locale :	50
2.14	Recherche Tabou :	51
2.15	Recuit simulé :	52
2.16	Algorithmes colonies de fourmis (ACF) :	54
2.17	Inspiration :	54
2.18	Métaheuristique :	55
2.19	Conclusion :	59
3	Chapitre 3 : élaboration du modèle et solution proposée	60
3.1	Introduction :	61
3.2	Classification et caractéristiques du problème :	61
3.3	Modélisation du problème :	62
3.4	Préparation des données :	73
3.5	Limites du modèle :	79
3.6	Perspectives futures :	80
3.7	Conclusion :	81
	Conclusion générale :	82
	Bibliographie :	83
	Annexes :	85

Liste des tableaux

Tableau 1 Explication du rôle réalisé par chaque division.....	20
Tableau 2 Analyse externe à l'aide des forces de Porter	26
Tableau 3 Diagramme SWOT qui montre les forces et faiblesses de Yassir Algérie ..	27
Tableau 4 résumé du cadrage de l'analyse stratégique sur Yassir Algérie	27
Tableau 5 Résumé de la nomenclature des problèmes dérivés du VRP.....	41
Tableau 6 Récapitulation des caractéristiques des VRP auxiliaires	43
Tableau 7 Les valeurs suggéré pour l'ACF.....	58
Tableau 8 Résumé des caractéristiques du probleme VRP considiré.....	62
Tableau 9 Articles de recherche abordant le problème de livraison.....	64
Tableau 10 Description des différentes constantes de l'algorithme modifié	68
Tableau 11 description de plusieurs solveurs d'optimisation	72
Tableau 12 Matrices des distances du problème a resoudre.....	73
Tableau 13 Les constantes simulés durant la seconde etude	77

Liste des figures

Figure 1 Liste des plateformes reconnu à l'échelle internationale.....	13
Figure 2 La différence entre GitHub (A gauche) et Git (à droite).....	15
Figure 3 La différence entre une entreprise plateforme (droite) et une entreprise de production (gauche)(Chou et al., 2017).....	15
Figure 4 Chiper, Opay, Chefaa et Jumia , quatre plateformes qui dominent l'Afrique	17
Figure 5 Organigramme de Yassir Algérie	18
Figure 6 Présentation du contenu de l'application d'un restaurant, réalisé par l'équipe Content de Yassir Market.....	22
Figure 7 Cartographie du macro processus service livraison de Yassir Express	28
Figure 8 Processus de gestion de la relation client.....	29
Figure 9 diagramme BPMN du processus de gestion de la relation client.....	30
Figure 10 Processus de gestion de la relation partenaires	30
Figure 11 diagramme BPMN de gestion relation partenaires	31
Figure 12 Processus de gestion de la livraison	32
Figure 13 diagramme BPMN montrant la gestion de la livraison.....	32
Figure 14 Diagramme BPMN global du service livraison de Yassir Express	34
Figure 15 Les différents types de ramassage en PDP(Toth and Vigo, 2014).....	44
Figure 16 Diagramme qui résume l'approche Branch and Bound appliqué au problème considéré(Casquilho, n.d.).....	48
Figure 17 Résumé de l'approche recherche tabou	52
Figure 18 Résumé du recuit simulé	54
Figure 19 L'expérience des doubles ponts des colonies de fourmis.....	55
Figure 20 Les caractéristiques des problèmes one-to-one(Toth and Vigo, 2014).....	62
Figure 22 A gauche l'image représente la situation du système de livraison avant la réception d'une nouvelle commande. A droite, Lors de la réception, l'algorithme calcule le chemin optimal de chaque livreur	66
Figure 23 l'affectation de la nouvelle commande	66

Figure 24 Pseudo code de l'ACF modifié.....	69
Figure 25 Methode de génération d'une route sur Python	74
Figure 26 La méthode qui résout le sous-problème	75
Figure 27 Insertion des caractéristiques sur Python	75
Figure 28 Résultat obtenu.....	75
Figure 29 convergence de la solution optimale pour plusieurs itérations de l'algorithme	77
Figure 30 Résultat des couts de transport sur la variation de n_best pour la première simulation	78
Figure 31 Résultat des couts de transport sur la variation de n_best pour la seconde simulation	78

Liste des abréviations :

VRP	Vehicle Routing Problem
CVRP	Capacitated Vehicle Routing Problem
PDP	Pickup and Delivery Problem
PDPTW	Pickup and Delivery Problem with Time Windows
DVRP	Dynamic Vehicle Routing Problem
TSP	Traveling Salesman Problem
TSPPD	Traveling Salesman Problem with Pickup and Deliveries
VRPTW	Vehicle Routing Problem with Time Windows
ACF	Algorithme Colonies de Fourmis
BPMN	Business Process Management Model
NP	Non-deterministic Polynomial Time
P	Polynomial Time
D-PDPTW	Dynamic Pickup and Delivery Problem with Time Windows
GVRP	Generalized vehicle routing problem

Introduction générale :

L'environnement du service de livraison à domicile en Algérie a connu une forte croissance durant les dernières années. Les entreprises qui vont détenir le marché sont les seuls qui pourront subsister durant les prochaines années. Face à cette concurrence féroce entre les entreprises, ces dernières redoublent d'effort et créativité pour demeurer plus concurrentiel.

Face à ces réalités, Yassir souhaite automatiser une grande partie de son service de livraison, cette automatisation permettra de réduire les coûts de livraison des biens et ainsi maximiser la satisfaction des clients et partenaires de l'entreprise.

Ce projet est un véritable challenge pour une startup encore en développement. La livraison des clients de Yassir doit se faire dans une durée limitée, même lorsqu'il existe des demandes de livraison importantes.

La satisfaction des clients n'est pas le seul paramètre à satisfaire. Chaque livreur partenaires de Yassir doit aussi maximiser son bénéfice durant sa disponibilité. Cette satisfaction aide Yassir à attirer plus de livreur et ainsi satisfaire plus de client. C'est un cercle vertueux.

L'affectation automatique des commandes diminue les coûts de livraisons du service de Yassir et maximise la satisfaction des partenaires. C'est dans cette optique que l'étude est réalisée.

Le premier chapitre présente le secteur de l'entreprise. En premier lieu, une description détaillée des plateformes entreprises est réalisée. La seconde partie présente l'entreprise Yassir et son contexte de développement. La dernière partie présente un diagnostic stratégique et global réalisé pour cerner la problématique du projet et de tirer les concepts importants à définir.

Le deuxième chapitre est une revue théorique des différentes méthodes et définition importantes pour cerner les éléments de la problématique. Plusieurs modèles mathématiques sont décrits pour résoudre les problèmes de tournées de véhicules. A la fin, plusieurs méthodes de résolution des problèmes d'optimisation mathématiques sont décrites, à savoir des méthodes de résolution exacts et approximatives.

Le dernier chapitre est consacré au développement de la solution. En se basant sur les concepts définis en second chapitre, la catégorisation du problème est réalisée, suivi d'une approche de résolution au problème ; Il s'agit d'utiliser un algorithme de colonie de fourmis pour résoudre le problème d'affectation des livreurs. Une partie de cette approche de résolution est implémentée et optimisé. A la fin du chapitre, plusieurs améliorations et suggestions sont décrites pour permettre la réalisation du projet.

Pour clôturer ce travail, une conclusion récapitule les étapes suivies et met en œuvre son apport et perspectives d'améliorations futures.

1 Économie des plateformes et son évolution en Algérie

1.1 Introduction

Ce premier chapitre introduit l'économie des plateformes, leur évolution dans l'ère du numérique ainsi que les acteurs principaux dans le monde. Dans une seconde partie, un acteur des plateformes en Algérie est présenté, Yassir, et l'organisation de son environnement.

1.2 L'économie des plateformes :

A cause d'une meilleure pénétration d'internet en Algérie, plusieurs applications émergent. Ces applications qui facilitent le quotidien des algériens n'est pas un phénomène unique en Algérie. Plusieurs pays ont subi cette évolution bien avant l'Algérie, cela dit, l'Algérie n'est pas épargnée de l'économie des plateformes.

1.2.1 Définition d'une plateforme :

Une plateforme est un intermédiaire qui permet la collaboration entre des collaborateurs (utilisateurs et producteur) de réaliser plusieurs types d'activité à l'aide de normes bien établies en créant de la valeur ajoutée(Moazed and Johnson, 2016).

Pour une définition plus simple, une plateforme d'échange de valeur entre plusieurs utilisateurs, par exemple un producteur et un consommateur. Cette plateforme n'est pas nécessairement digitale, par exemple un agent immobilier facilite à un vendeur ou locataire de bien immobilier de trouver un client intéressé et vice versa. L'agent immobilier dans notre cas joue le rôle d'une plateforme.

Par contre, les plateformes numériques sont devenues de très grande entreprise par rapport au plateforme simple. A l'aide d'internet, YouTube est devenu un lien entre des producteurs de contenu numérique et des utilisateurs intéressés, le producteur gagne un pourcentage monétaire des publicités publiés avant la visualisation de ses vidéos et le consommateur a un accès à un programme qui l'intéresse. La figure ci-dessous montre plusieurs services qui sont considérés comme des plateformes. La majorité de ces services sont le produit d'entreprise connu globalement (Amazon, google...)



Figure 1 Liste des plateformes reconnu à l'échelle internationale

Une caractéristique principale des plateformes est la non-acquisition des services réalisés par les producteurs. YouTube n'a pas de droit d'auteur sur les vidéos publiés, Uber n'a pas le véhicule du chauffeur. Ces plateformes font seulement la liaison avec les clients.

1.2.2 L'impact des plateformes : le cas de GitHub :

Plusieurs personnes pensent que la valeur ajoutée de ces plateformes numérique est seulement dans la facilité de communication. Or en réalité, ces plateformes peuvent être la solution à un problème, ce qui est le cas de GitHub(Moazed and Johnson, 2016).

En 2005, Linus Torvalds a créé le système Git. Ce système a pour but de faciliter la maintenance de code informatique entre plusieurs contributeurs. Avant ce système, pour maintenir les fichiers et le code informatique d'un programme, les programmeurs modifient directement le code qui se trouve dans l'application finale. Cette approche cause plusieurs problèmes, particulièrement le risque de perdre les versions précédentes du logiciel si on fait une mauvaise modification. À l'aide de Git par contre, chaque programmeur peut créer une copie du code principal, modifier et vérifier le bon fonctionnement du logiciel, puis notifier les autres contributeurs, si ces contributeurs acceptent la modification, le code principal est ainsi modifié.

Cette approche a permis à Linus Torvalds d'implémenter plus rapidement les modifications du code sur son projet open source Linux.

GitHub a démocratisé l'utilisation de Git. En donnant une interface graphique au lieu d'une interface en ligne de commande. Cette interface graphique facilite la communication entre les collaborateurs et aide les utilisateurs à mieux cerner les changements. La figure 2 donne une comparaison entre Github et Git. Clairement GitHub semble plus intuitive, on n'a pas besoin de connaître le nom d'une commande pour l'utiliser.

GitHub, une plateforme de programmeur, a changé le monde de la programmation. L'utilisation de GitHub est gratuite, mais pour stocker votre logiciel en mode privé, c'est-à-dire seul quelques utilisateurs peuvent accéder au code, c'est payant. Plusieurs entreprises utilisent GitHub pour gérer leur logiciel, par exemple Microsoft.

Actuellement, la majorité des collaborations entre programmeur utilise GitHub pour développer des logiciels. GitHub est seulement un exemple parmi tant d'autre qui représentent l'importance de la facilité de communication et l'importance de la simplification, deux critères contribuant à la réussite des plateformes connues.

1.2.3 Entre une entreprise plateforme et une entreprise de production :

Les plateformes représentent un nouveau business model par rapport au modèle de production. La figure 2 représente un schéma d'une entreprise de production de bien matériel(usine) et en parallèle la démarche d'une entreprise plateforme.

Les entreprises de production ont un système de création de valeur linéaire. L'entreprise de production reçoit la matière première d'un prestataire, puis elle fait une transformation et à la fin un client utilise le produit. L'entreprise de production est chargée de faire le marketing, assurer une meilleure prestation de services...

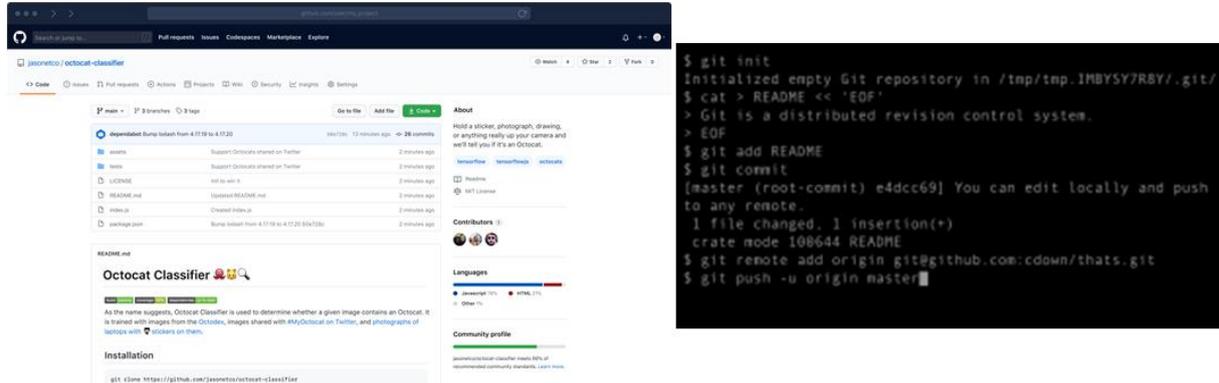


Figure 2 Une comparaison entre l'interface de Github et GIT

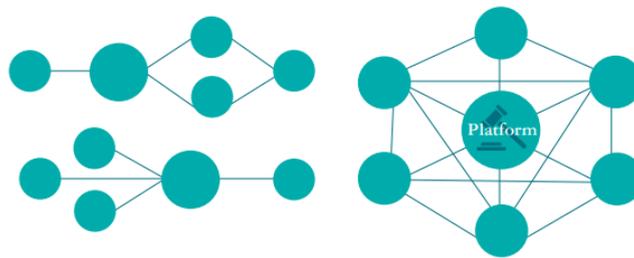


Figure 3 La différence entre une entreprise plateforme (droite) et une entreprise de production (gauche)(Chou et al., 2017)

Par contre, Une entreprise plateforme joue le rôle d'intermédiaire entre le producteur et le consommateur. Elle assure une facilité de communication entre les demandeurs de services et producteur, l'équilibre entre la demande et la prestation est aussi assuré par l'entreprise. Par exemple, Uber, une startup spécialisée dans le secteur du VTC (Voiture de Transport avec chauffeur), régularise la demande en mettant des tarifs particuliers pour les trajets de nuit, cette tarification pousse les chauffeurs à rester connecté durant la nuit et donc satisfaire plus de client lors des weekends, par exemple.

1.2.4 Facteur clés du succès des plateformes :

Ce nouveau business model a bouleversé l'économie mondiale, le monde c'est transformé d'une économie accès production a une économie accès prestation de service.

Selon Deloitte(Chan et al., 2018), il existe plus de 187 plateformes d'une évaluation de plus de 1 milliards de dollars. Ces startups sont particulièrement concentrées en Amérique du Nord (États-Unis) et en Asie (Chine).

Il existe plusieurs facteurs qui ont permis l'émergence de ce modèle en Chine et aux États-Unis. On peut citer en particulier :

- **Présence d'une population technophile** : Aux États-Unis et en Chine, la présence d'une population importante, et en particulier une population qui utilise l'internet quotidiennement, a permis l'adoption rapide des plateformes. La génération Z est le marché des plateformes.
- **Absence de régulation** : Dans plusieurs pays, le marché touché par les plateformes n'était pas régulé jusqu'à l'émergence de ces dernières. Uber et Meituan en ont profité dans leurs marchés locaux pour se développer, et cette absence de régulation a permis une domination globale du marché. Une domination non-apprécié par les gouvernements.
- **Présence d'une culture d'innovation** : Les plateformes, particulièrement de la dernière décennie, sont le produit de plusieurs expériences entrepreneuriales en Silicon Valley et Shenzhen. Ces hubs des startups connues à travers le monde (Facebook, Google, Tencent...) contiennent des centaines d'investisseurs particulier et Capitals risques qui ont l'habitude d'investir dans des entreprises risquées. D'où l'émergence de ce nouveau modèle.
- **Économie d'échelle** : les plateformes permettent une meilleure tarification à l'aide de la digitalisation. Cette approche a permis une réduction des prix de livraison sur UberEats par exemple.

1.2.5 Les plateformes en Afrique :

La figure 3 montre quelques plateformes qui ont pu dominer le marché africain. Même si le marché africain a connu un retard dans le domaine des plateformes. La seule entreprise avec évaluation de plus d'un milliard de dollars est l'entreprise Naspers. Récemment, plusieurs startups ont émergé dans le marché africain, par exemple Jumia se spécialise dans le marché e-commerce africain.

Ces startups bénéficient d'un investissement étranger majeur. En 2019, le volume monétaire des levés de fonds a augmenté de plus de 74% par rapport à 2018, dépassant ainsi la barre des 2 milliards de dollars (*E-conomy Africa 2020*, 2020). La majorité de ces investissements sont concentré sur le domaine de la Fin Tech et l'E-commerce, c'est-à-dire des plateformes entreprises.

En Algérie, plusieurs entreprises existent. L'une des premières plateformes en Algérie est probablement OuedKniss, mais depuis 2016, il existe plusieurs acteurs dans le domaine on peut citer Yassir, TemTem et Jumia.

1.3 Yassir : une plateforme Algérienne :

Yassir est une startup algérienne fondé en 2017 par Al-Mahdi YETTOU, Noureddine Tayebi et Mustapha Baha. L'entreprise fait partie des premières startups en Algérie, et parmi

les premières startups financées en Algérie par l'entreprise de financement précoce Y Combinator.

Le mot « Yassir » est un mot arabe qui signifie facile ou aisé. Yassir a pour but de faciliter le quotidien des algériens, d'où le nom Yassir.



Figure 4 Chiper, Opay, Chefaa et Jumia¹ , quatre plateformes qui dominent l'Afrique

Actuellement, Yassir est présente dans quatre pays qui sont l'Algérie, la France, le Maroc et la Tunisie. L'entreprise est accès sur trois secteurs d'activité :

- Service VTC (Voiture de Transport avec Chauffeur)
- Service livraison et restauration (Yassir Express)
- Service E-commerce (Yassir Market)

Yassir est parmi les leaders en Algérie dans le service VTC. Elle compte plus de 40.000 partenaires et propose ses services dans 25 villes et a plus de 2 millions de prestation satisfaite(Saidoun, 2020).

¹ Chiper est une entreprise nigérienne qui permet de transférer l'argent entre les pays africains a un cout très faible, le transfert d'argent entre les citoyens du même pays est gratuit. Opay est aussi une Fintech spécialisé dans le service de e-paiement. Chefaa est une startup égyptienne qui permet la livraison des traitements médicaux.

1.4 Activité et organisation de Yassir en Algérie :

Le service de Yassir Algérie est disponible dans plus de 10 villes en Algérie. Yassir Algérie détient l'intégralité du processus de conception (développement) jusqu'au lancement de l'application. Les services disponibles en Algérie sont :

- Service VTC : Ce service représente l'activité principale de l'entreprise en Algérie. Cette activité a commencé depuis la création de l'entreprise en 2016. Le service assuré est un service de demande de transport d'un lieu à un autre à l'aide d'une application mobile.
- Service Express (Yassir Express) : Ce service représente l'activité secondaire de l'entreprise en Algérie qui a commencé en 2019. Le service express aide les clients à avoir les plats de leur meilleur restaurant dans le secteur du client.
- Service Market : Ce service a commencé fin 2020, il a pour but de créer un marché E-commerce algérien, comme OuedKniss et Jumia Market.

Il existe des bureaux à travers les pays où Yassir est disponible. Ces bureaux sont chargés du marketing des services et de la collaboration avec les restaurants et chauffeurs disponibles dans le pays.

Le schéma suivant représente l'organigramme de l'entreprise, en détaillant les structures les plus importantes dans mon projet:

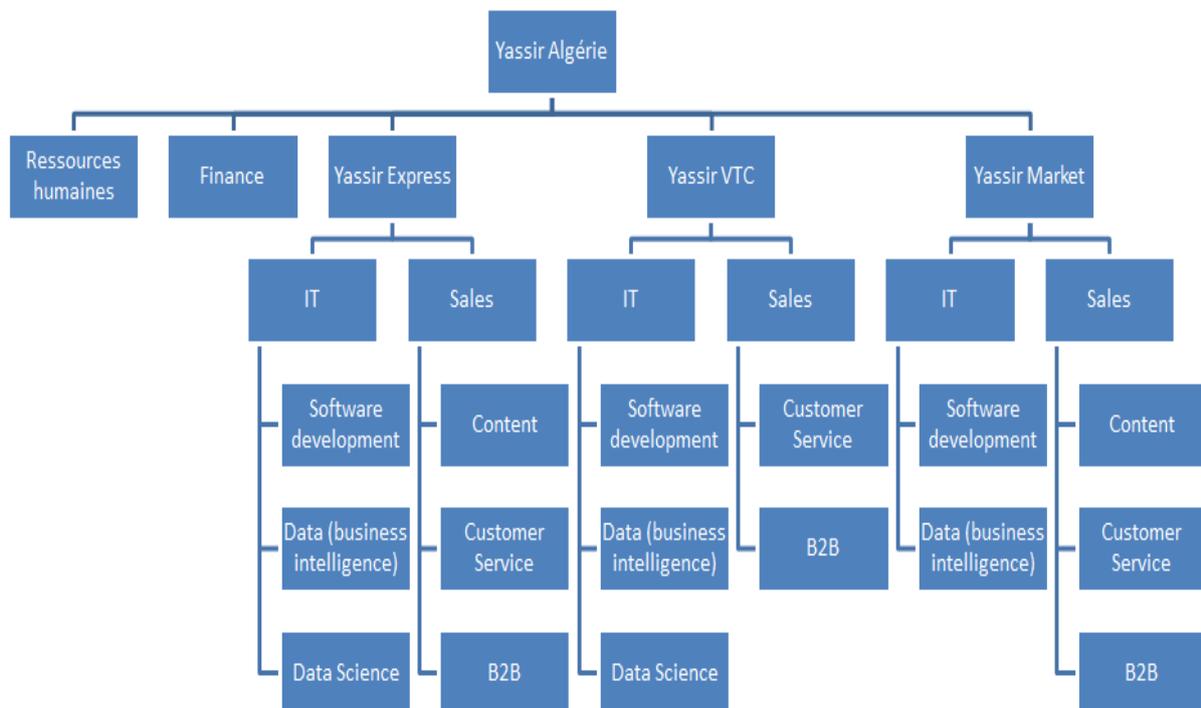


Figure 5 Organigramme de Yassir Algérie

On remarque que la structure de l'entreprise est divisionnelle. Chaque partie de l'entreprise est spécialisée dans une fonction déterminée, Le responsable du département Data de Yassir Express gère la fonction Data de cette partie de l'entreprise, de même pour les autres départements.

Je me suis intéressé au fonctionnement de deux départements principalement, et ces deux départements sont le département Sales et le département IT. Car ces les départements qui ont le rôle le plus important dans le projet.

1.4.1 La division Sales :

La division Sales est responsable de la relation avec les différents prestataires externes. La division compte 3 départements qui gèrent la relation avec les prestataires externes de chaque service.

Particulièrement, Chaque département contient plusieurs équipes qui sont responsables de trouver des nouveaux partenaires pour développer les services de Yassir (marketing développement), gérer le contenu de ce partenaire dans l'application de Yassir et assurer la relation client avec le client final. Ces tâches diffèrent d'un Service à l'autre. Parfois un département peut ne pas réaliser une des tâches, car cette tâche ne rentre pas dans le besoin de la fonctionnalité de l'application finale. Le tableau 1 montre si un service nécessite une tâche et une explication détaillée sur le rôle de cette dernière.

Le plus important dans ce diagramme c'est le rôle que réalise l'équipe de Yassir Food, car cette partie est la plus importante dans notre problématique.

1.4.1.1 L'équipe Sales de Yassir Express :

L'équipe Sales de Yassir Market est chargée de gérer l'intégralité des relations externes avec les parties prenantes qui sont :

- Les livreurs contractés par Yassir pour assurer la livraison des clients.
- Les vendeurs partenaires de Yassir.
- Les clients qui prennent des commandes de Yassir.

Ces parties prenantes, particulièrement les vendeurs et les livreurs, doivent assurer une satisfaction maximale des clients. Cette Satisfaction est mesurée à travers les critères suivants:

- La satisfaction des commandes demandées par le client :
 - Chaque produit mentionné dans l'application doit être disponible dans l'inventaire du vendeur choisi par le client. Si un produit n'est pas disponible dans l'inventaire, l'application de Yassir Market doit montrer l'indisponibilité du produit.

Tableau 1 Explication du rôle réalisé par chaque division du membres sales

	Gérer la relation avec le client final	Gérer le contenu de l'application	Attirer des nouveaux prestataires pour le service
Yassir VTC	L'équipe relation client de Yassir VTC doit s'assurer qu'un client est servi par un chauffeur et livré à sa destination désiré	La fonction de la gestion de contenu est affectée au département IT	Cette fonction est gérée par l'équipe de recrutement, qui a pour but d'attirer de nouveau chauffeur au service
Yassir Express	S'assurer qu'une commande client (repas) est satisfaite, du contact des restaurants a la livraison du chauffeur	L'application contient plusieurs restaurants, avec leur plat, images Ce contenu est géré par les CS de Yassir Express	Les CS de Yassir Express sont responsables seulement d'attirer de nouveau restaurant, et non des chauffeurs
Yassir Market	Cette équipe s'assure de la livraison du produit a temps au client finale	L'application contient plusieurs produits, avec leurs images Ce contenu est géré par les membres relation clients de Yassir Market.	Cette fonction est gérée par l'équipe de recrutement, qui a pour but d'attirer de nouveau chauffeur au service

- Le temps de livraison des commandes :
 - Yassir Market suggère un temps de livraison pour satisfaire les clients. Un temps supérieur à cette recommandation pénalise le développement du service de livraison, car une partie importante des clients provient des recommandations d'utilisateurs de l'application.
- La partie relationnelle avec le client :
 - Les livreurs et les membres du service relation client de Yassir doivent avoir un excellent relationnel avec le client. Plusieurs problèmes peuvent arriver

lors de la livraison, un livreur ou un sales excellent en relationnel peut garder le client satisfait. De plus, Yassir sélectionne les livreurs et sales pour s'assurer de la bonne conduite de ces représentants ; les clients peuvent toujours contacter Yassir en cas de problèmes.

Pour assurer la satisfaction des clients, l'équipe Sales de Yassir suit l'intégralité de « l'expérience client » c'est-à-dire l'expérience du client sur l'application de Yassir Market. Pour cela l'équipe Sales de Yassir Market est constituée de trois équipes :

1.4.1.1.1 L'équipe contenu (Content) :

L'équipe contenu (Content) gère les interfaces des vendeurs qui sont disponibles sur Yassir Market. S'il y a un nouveau restaurant qui a rejoint l'application, l'équipe Content crée le contenu de ce vendeur, c'est-à-dire la liste des produits qui vend sur l'application. L'équipe Content s'assure aussi de modifier le contenu des restaurants existants sur l'application de Yassir. La figure 6 montre une page qui a été réalisée par l'équipe.

1.4.1.1.2 L'équipe B2B :

L'équipe B2B de Yassir Market assure la relation avec les vendeurs. Les magasins, nommé aussi partenaires, sont en général contacté par l'équipe B2B pour proposer le service de livraison de Yassir Market. Si le partenaire accepte les termes de Yassir, Le vendeur est ajouté à l'application. L'équipe B2B assure non seulement la sélection de nouveau vendeur, mais aussi le suivi de la relation client.

1.4.1.1.3 L'équipe relation client (Customer Service) :

Cette équipe gère la relation client particulièrement :

- Communiquer avec le client pour s'assurer de sa localisation et de rassurer le client, en cas de retard de livraison.
- S'assurer que les vendeurs traitent la commande des clients.
- Trouver un chauffeur qui va transporter la commande du magasin jusqu'au client.
- Communiquer avec le chauffeur pour s'assurer de la livraison du produit.

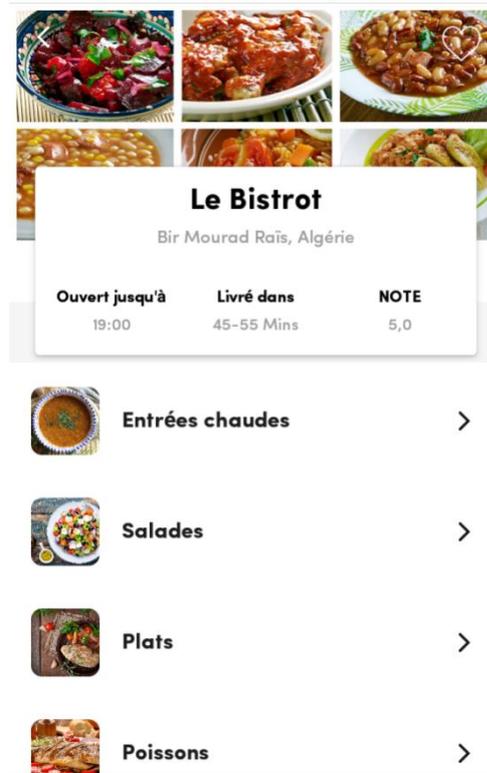


Figure 6 Présentation du contenu de l'application d'un restaurant, réalisé par l'équipe Content de Yassir Market

1.4.2 La division IT :

La division IT de Yassir Algérie est chargée de maintenir l'intégralité des services IT de Yassir à jour. La division est composée de plusieurs départements qui sont :

1.4.2.1 Le département software développement :

Ce département est chargé de la conception, maintenance et améliorations de l'intégralité des applications de Yassir. Ce département assure aussi le bon fonctionnement des besoins interne en application. Par exemple L'équipe Content décrite précédemment utilise un tableau de bord qui les aident à réaliser les modifications nécessaires sur l'application de Yassir Express.

1.4.2.2 Le département Business Intelligence (BI) :

Ce département est chargé de la conception du tableau de bord pour l'utilisation par les différents départements de l'entreprise. Par exemple, la réalisation d'un tableau de bord pour l'équipe RH.

1.4.2.3 Le département Data Science :

Ce département est chargé d'analyser les données tirées des utilisateurs des applications de Yassir pour trouver des améliorations. Par exemple, étudier le déplacement des clients

permet une meilleure recommandation d'emplacement du chauffeur. Ce département est important pour l'entreprise car, souvent, il permet une meilleure connaissance des améliorations nécessaires.

1.5 Diagnostic du marché de l'entreprise :

Dans un environnement changeant et plein de compétitivité, un diagnostic stratégique est nécessaire pour déterminer les forces et les faiblesses de l'entreprise. Un diagnostic nous permet aussi de connaître le contexte où l'entreprise évolue ainsi que les approches qui doivent être prises pour assurer la pérennité de l'entreprise.

De ce fait, un diagnostic stratégique est réalisé pour déterminer la stratégie de Yassir et les risques qui peuvent impacter la survie de l'entreprise (Brulhart and Safari, 2011). Dans la deuxième section et à l'aide de l'étude de la première partie, un diagnostic externe est réalisé suivant l'approche des cinq forces de Porter et le diagramme SWOT. Enfin, un diagnostic interne est réalisé sur la fonction Yassir Express à l'aide de l'approche processus, car cette fonction est la plus importante pour tirer la problématique.

1.6 Diagnostic stratégique des services de Yassir :

1.6.1 Diagnostic service « Yassir VTC » :

L'application Yassir VTC est le premier service introduit par l'entreprise. Depuis 2016 et jusqu'à 2019, une forte croissance a été enregistrée dans ce secteur d'activité de l'entreprise. Même si les chiffres sont rares sur la taille de Yassir sur le secteur VTC. L'entreprise estime qu'il y a plus de 2 millions d'utilisateurs et 40.000 prestataires de service (Saidoun, 2020). Yassir a confirmé sa position, leader du marché VTC Algérien. A cause de cela Yassir a essayé de conquérir de nouveaux marchés, notamment le marché Marocain et Tunisien.

Par contre, depuis 2019, de nouveaux prestataires de service similaires ont émergé dans le marché maghrébin. Ces entreprises financées de l'étranger ont une tendance baissière des prix et ont devenu un vrai compétiteur à l'application. De plus, la pandémie a affecté négativement l'activité de ce secteur à cause du couvre-feu instauré par les gouvernements de la région. Yassir est effectivement affecté par cette pandémie, mais jusqu'à maintenant, à cause d'une bonne gestion financière avant la pandémie, l'entreprise a pu survivre durant la crise.

1.6.2 Diagnostic service « Yassir Express » :

Le service de livraison Yassir Express a été introduit en 2019. L'activité est en train de grandir même si d'autres entreprises existaient déjà bien avant l'introduction du service à Yassir. Plusieurs raisons suggèrent que le service va grandir et même avoir la même importance que le secteur VTC :

- Yassir essaye de maximiser le nombre de partenaires présents sur l'application, pour satisfaire au maximum les besoins du client.

- Les acteurs présents sur le marché (Jumia, Food delivery...) n'ont pas pu dominer le marché à cause de la pandémie.

Même si Yassir a rejoint ce marché en retard, ce marché de niche a un fort potentiel de développement dans les années à venir.

1.6.3 Diagnostic service « Yassir Market » :

Le service Market est le service le plus récent des services de Yassir. Même si le projet est nouveau, la situation du marché algérien montre un vrai besoin d'un nouveau site E-commerce Algérien :

- Même si Ouedkniss continue d'avoir une présence majeure sur le quotidien algérien, l'interface non évolutive de l'application agace les utilisateurs. Yassir Market avec une interface plus intuitive l'aidera à dominer le marché.
- Yassir compte éviter les mauvaises pratiques utilisées par d'autres site E-commerce, notamment la mauvaise publicité de promotion et les longues durées de livraison. Yassir inspire confiance au partenaire et utilisateur du site à cause de son excellent service VTC.

Si Yassir garde un excellent relationnel avec les différentes parties prenantes, ce service a le potentiel de remplacer le service VTC en importance.

1.7 Analyse externe suivant les cinq forces de Porter :

A l'aide des observations décrites précédemment, le tableau 2 montre les observations constatées sur le marché de Yassir à l'aide du diagramme de Porter.

Le résultat de l'analyse de l'environnement de l'entreprise nous permet de comprendre les mécanismes et les évolutions des marchés ciblés par Yassir. Le marché est caractérisé par une forte concurrence et un fort pouvoir de négociation des clients et fournisseur. De plus, la présence de plusieurs firmes rendent le marché très concurrentiel.

Ces caractéristiques du marché justifient la stratégie de Yassir. Mise à jour continue des prix et recrutement continu de fournisseurs permettent à l'entreprise de développer son marché.

1.7.1 Analyse stratégique suivant la matrice SWOT:

La matrice SWOT, ou *Strength, Weaknesses, Opportunities and Threats* est un outil d'analyse stratégique qui permet à une entreprise d'évaluer son positionnement stratégique par rapport à plusieurs facteurs (internes et externes) qui affectent l'entreprise.

Les résultats de l'analyse sont résumés dans le tableau 3.

1.7.2 Résultat de l'analyse stratégique :

Les résultats de l'analyse stratégique de Yassir nous permettent de comprendre l'évolution du marché algérien du VTC ainsi que la stratégie adoptée par Yassir. En résumé, les marchés où Yassir est présente ont connu une forte compétitivité durant les dernières années ; Ces marchés ont aussi les caractéristiques d'aisance de basculement des fournisseurs et clients d'un prestataire de services à un autre. De plus, puisque ces marchés sont nouveaux, aucun prestataire ne peut prétendre dominer le marché d'une manière pérenne.

En parallèle, Yassir vise à avoir une plus grande part de marché en recrutant plus de fournisseurs, car avoir une grande sélection de fournisseur répartis à travers le pays aide à satisfaire plus de villes où, potentiellement, se trouvent les clients.

1.8 Analyse interne par l'approche processus :

A cause de la similitude des processus réalisés sur les différents services de Yassir. L'analyse des processus sera réalisée principalement sur le service « Yassir Express ».

En première partie, une identification des processus est réalisée pour déterminer les processus clés du service Yassir Express. Ensuite, une modélisation des processus clés est réalisée à l'aide de la méthode BPMN. Enfin, la détermination des dysfonctionnements constatés sur ce macro processus. En deuxième partie, les dysfonctionnements constatés sont cités et la problématique est décrite.

1.8.1 Cadrage de l'analyse :

Dans cette étape, on définit un périmètre de l'analyse des processus existants en entreprise, et particulièrement à Yassir Express. Ce cadrage permet d'étudier les flux logistiques gérés par Yassir Express.

Yassir Express est présente sur plusieurs villes en Algérie. Même si chaque ville est gérée séparément par un système de zonage dédié. Les processus constatés sont similaires. De ce fait, l'étude menée se focalisera principalement sur les services de Yassir Express à Alger. Le tableau 2 résume le périmètre de notre analyse.

Tableau 2 Analyse externe à l'aide des forces de Porter

Force	Observations
Menace des nouveaux entrants.	<ul style="list-style-type: none"> • Présence de plusieurs multinationales dans le domaine (barrière à l'entrée) • Absence de petite entreprise
Menace des fournisseurs (restaurant et livreurs)	<ul style="list-style-type: none"> • Les entreprises avec les meilleurs tarifs pour les conducteurs attirent plus de livreurs et prestataires • Peu de fournisseurs (livreurs) dans certaine région avec une forte demande client. • Faible coût de transfert (restaurant) vers d'autre prestataire
Menaces du pouvoir de négociation des clients.	<ul style="list-style-type: none"> • Forts pouvoir de négociation • Faible coût de changement de service • Concentration des clients (Alger, Oran, Annaba...) • Existence de source de substitution
Menace des produits de substitution.	<ul style="list-style-type: none"> • L'entreprise représente un nouveau produit dans le secteur
Menace de l'intensité concurrentielle.	<ul style="list-style-type: none"> • Plusieurs concurrents (Careem, Heetch...) • Faible différenciation des services

Tableau 3 Diagramme SWOT qui montre les forces et faiblesses de Yassir Algérie

Forces	Faiblesses
<ul style="list-style-type: none"> • L'entreprise est Présente sur plusieurs villes en Algérie • Un nombre de fournisseur (chauffeur et restaurant) élevé • L'entreprise propose un Service diversifié pour satisfaire une large gamme de produit. 	<ul style="list-style-type: none"> • La non automatisation de plusieurs processus, particulièrement l'affectation des fournisseurs (chauffeurs) • Absence de normes à satisfaire par les fournisseurs (prestations de services) • Absence de suivi de performances des prestataires
Opportunités	Menaces
<ul style="list-style-type: none"> • Automatisation des affectations clients • Amélioration de la communication avec les fournisseurs. • Expansion sur le continent africain et l'Afrique francophone 	<ul style="list-style-type: none"> • Compétitivité élevée par des startups global (Careem, filiale de Uber ...) • Absence de frais de changement de prestataires pour les clients • Affaiblissement de la flotte partenaire de Yassir • La continuité de la pandémie sur les marchés visés par Yassir (Algérie)

Tableau 4 résumé du cadrage de l'analyse stratégique sur Yassir Algérie

Cadrage de l'analyse

Géographie	Alger
Produit	Yassir Express
Organisation	Livraison : service Sales

1.8.2 Identification des processus clés :

A partir du cadrage déjà réalisé sur la première étape. On peut définir l'ensemble des processus clés du périmètre. Ainsi que les processus de support et de pilotage :

- Processus de gestion relation client : une communication avec le client est nécessaire pour s'assurer de sa localisation et sa satisfaction.
- Processus de livraison : la livraison est réalisée par un chauffeur contracté avec Yassir.
- Processus de gestion relation partenaires : une communication avec les partenaires (restaurant ou magasin) est nécessaire pour s'assurer la disponibilité des produits commandés par le client.

La cartographie du macro-processus service de livraison représente l'ordre de ces processus dans l'activité de livraison à Yassir Express.

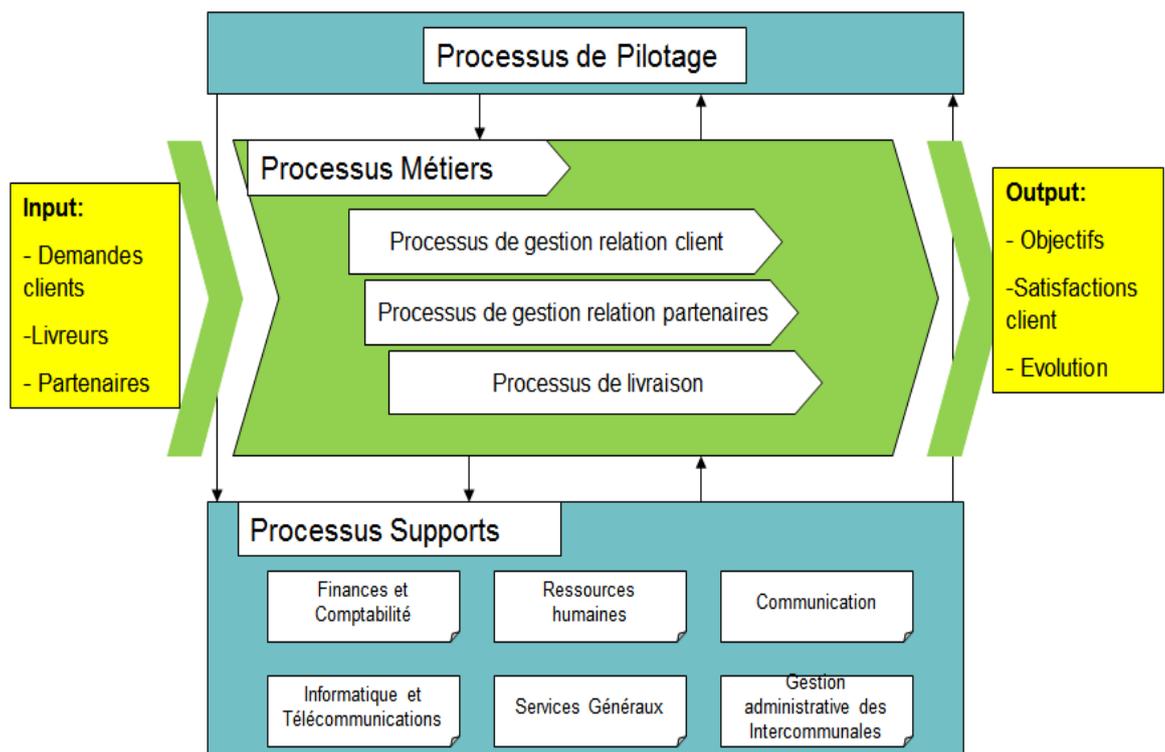


Figure 7 Cartographie du macro processus service livraison de Yassir Express

L'une des caractéristiques principales de ce processus est l'importance de la bonne gestion de l'intégralité de la procédure. Souvent les commandes client doivent être satisfaites dans une heure et donc les processus de gestion de la relation client et de gestion de la relation partenaire doivent se faire en quelques minutes.

1.8.3 Formalisation et description de chaque processus :

Dans cette partie, une description détaillée de chaque processus afin de déterminer l'utilité de ce dernier et les relations existantes avec chaque processus clés du service.

1.8.3.1 Processus de gestion relation client :

Ce processus peut intervenir plusieurs fois dans le macro-processus. Au début, il s'agit de contacter le client pour confirmer sa localisation. Ensuite, le client peut être contacté en cas d'annulation de sa commande. La commande peut être annulée pour plusieurs raisons, par exemple si le produit n'est pas disponible au restaurant ou bien la localisation du client est en dehors d'Alger. Enfin, le client peut avoir une réclamation à la fin d'une livraison, s'il n'est pas satisfait. La cartographie et le processus sont décrits dans la figure ci-dessus.

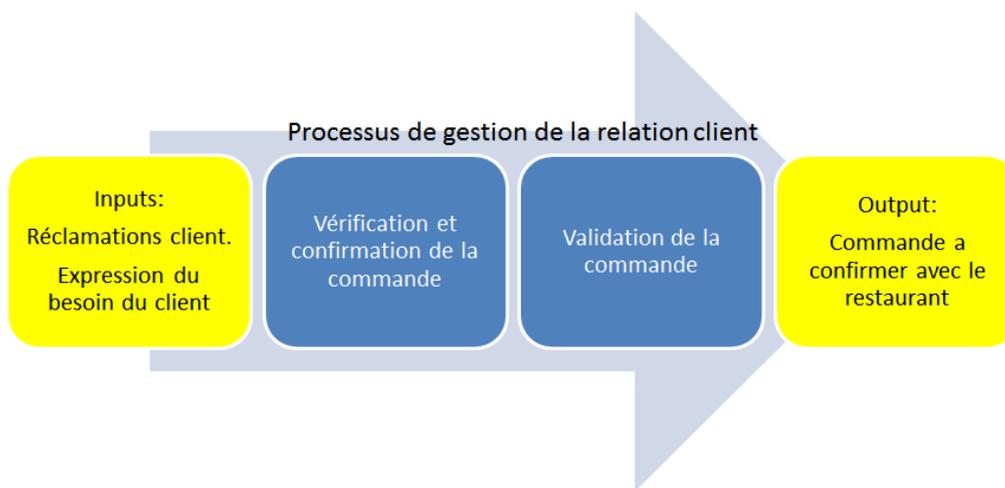


Figure 8 Processus de gestion de la relation client

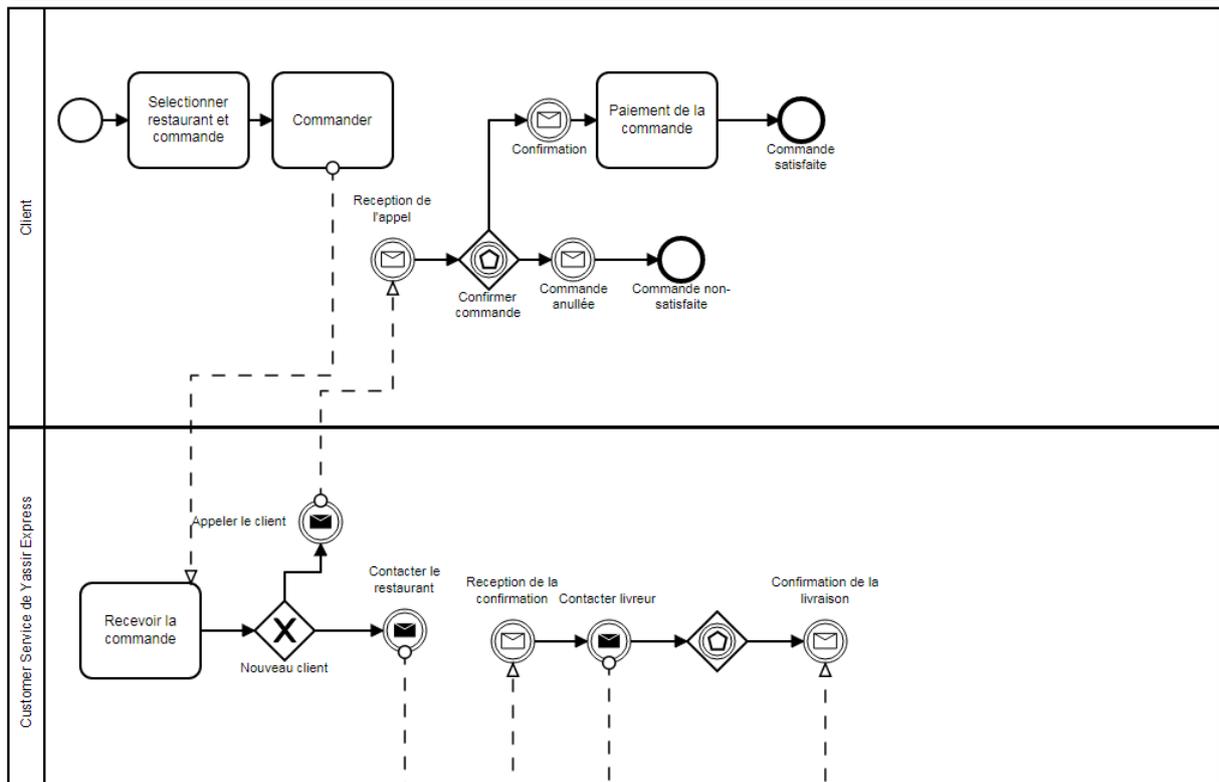


Figure 9 diagramme BPMN du processus de gestion de la relation client

1.8.3.2 Processus de gestion relation partenaire :

De même que la gestion relation client, ce processus peut intervenir sur plusieurs étapes. Le partenaire est contacté au début pour confirmer la disponibilité du produit. Ensuite pour confirmer la réception de la commande par le livreur. Le partenaire peut être contacté à la fin pour l'informer sur une réclamation client. La figure ci-dessus montre la cartographie du processus.

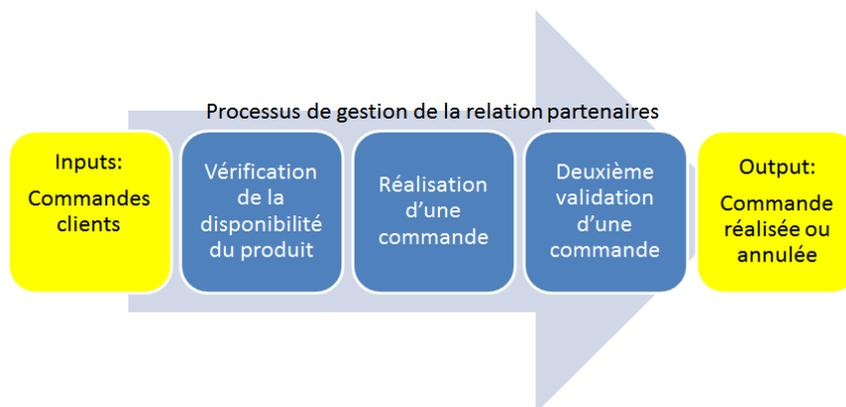


Figure 10 Processus de gestion de la relation partenaires

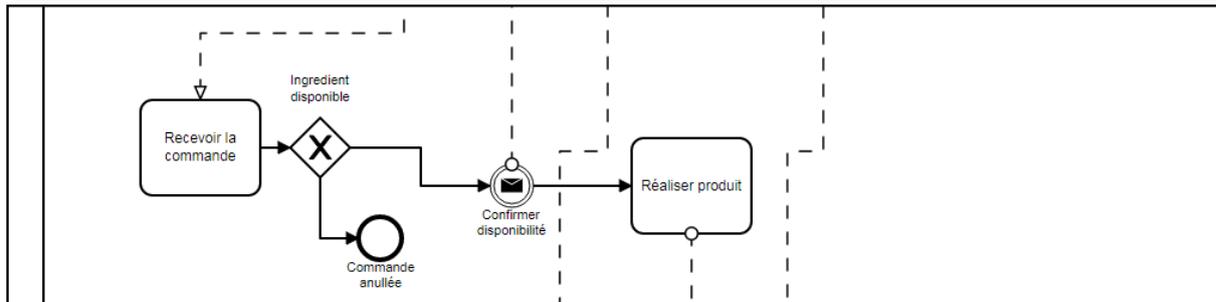


Figure 11 diagramme BPMN de gestion relation partenaires

1.8.3.3 Processus de livraison :

Les livreurs de Yassir Express sont localisés dans plusieurs régions à Alger. Chaque livreur possède un « box » qui protège le produit désiré des conditions externes (chaleur, humidité...)

Dans une première partie et après la validation de la commande client, l'équipe de Yassir Express contacte les livreurs qui sont les plus proches au vendeur et client. Si le livreur accepte de prendre la commande du client, le livreur se dirige vers le partenaire ou le client a fait sa commande et enfin le client est livré dans sa destination choisi. L'étape de sélection de livreur doit se faire rapidement, car l'un des critères de satisfaction du client est la courte durée de livraison.

Le livreur est aussi chargé de la réception du paiement du client. Premièrement, il paye le partenaire de Yassir les frais de la commande. Le client ensuite paye le livreur le prix de la commande et les frais de livraison. Ces frais de livraison représentent le bénéfice réalisé par le livreur dans chaque commande livrée.

Souvent, un livreur est affecté plusieurs commande client. Dans ce cas, le livreur peut accepter de prendre ces commandes mais il doit savoir les planifier, car l'application ne montre pas le chemin optimal lorsqu'il y a plusieurs commandes. La figure montre la cartographie du processus et sa représentation à l'aide d'un diagramme BPMN.

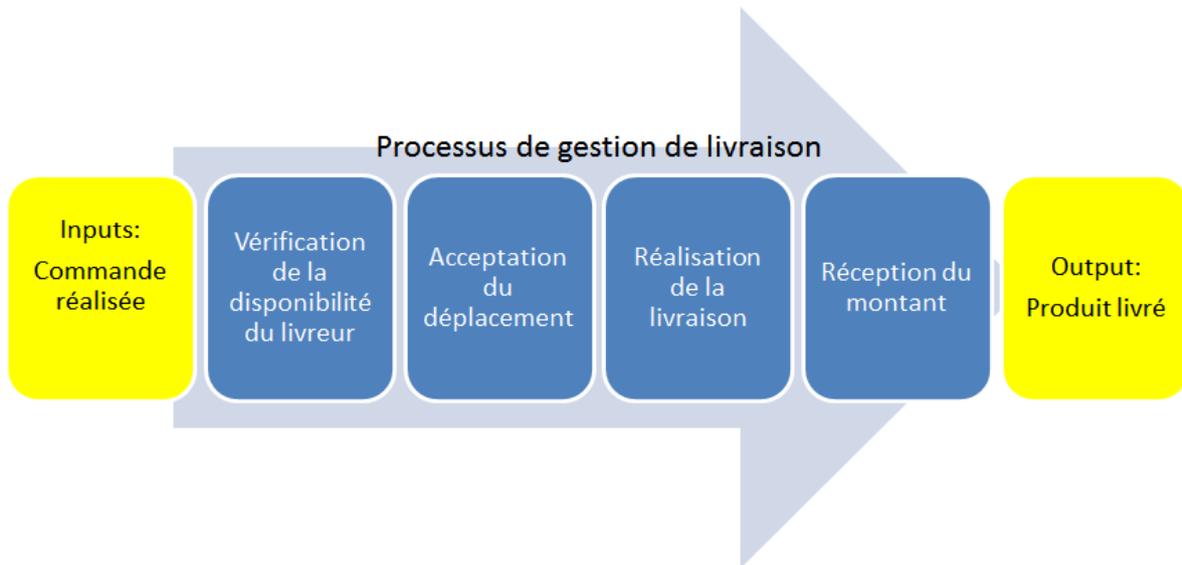


Figure 12 Processus de gestion de la livraison

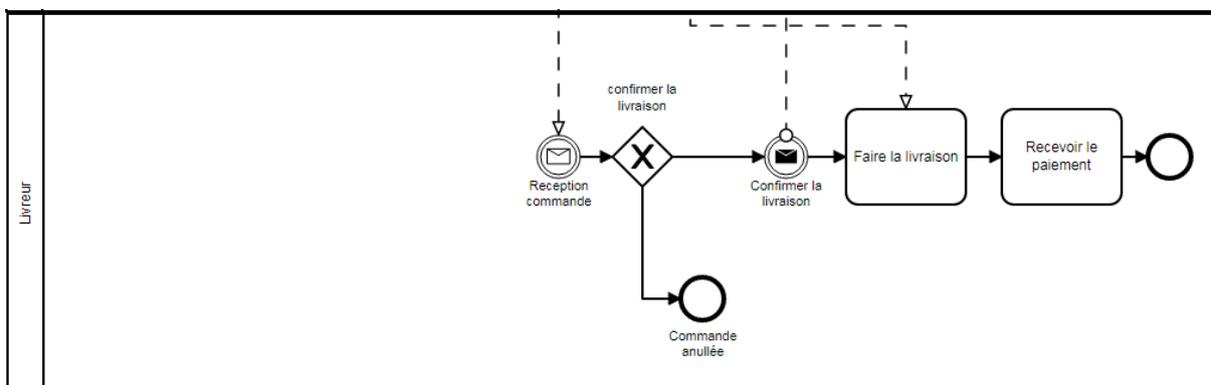


Figure 13 diagramme BPMN montrant la gestion de la livraison

1.8.3.4 Gestion des retours :

Les retours dans ce type de livraison est rare. Cependant, si un livreur n'arrive pas à trouver le client, le livreur est responsable de rendre la commande au partenaire de Yassir. Par exemple, si une commande d'un restaurant n'est pas livrée à un client parce qu'il ne répond pas, le livreur retourne la commande au restaurant et il est remboursé. Yassir Express s'en charge ensuite de payer le livreur les frais de livraison.

1.8.4 Elaboration d'une cartographie globale du macro-processus :

L'élaboration de la cartographie globale permet de montrer les liaisons du macro-processus de livraison de Yassir Algérie.

Après plusieurs observations du processus en action. Plusieurs dysfonctionnements sont constatés :

- **Dysfonctionnement 1** : Le processus d'affectation des commandes aux livreurs est complètement manuel. L'absence d'une procédure d'automatisation cause un temps important de livraison, car l'affectation manuelle nécessite un temps de communication important entre le livreur, partenaire et l'équipe Customer Service de Yassir Express. L'automatisation d'une partie de procédure est nécessaire pour économiser le temps de livraison.
- **Dysfonctionnement 2** : Absence d'un tableau de bord qui montre aux partenaires les commandes désirées par le client. Le restaurant reçoit souvent la commande de la part de l'équipe de Yassir Express, et cela augmente la possibilité d'une réalisation d'une mauvaise commande pour un client. Une application pour les restaurants est en cours d'élaboration pour permettre la visualisation des commandes au restaurant.

1.9 Résultat du diagnostic interne :

L'approche de Yassir express pour réaliser des demandes de livraison ne peut pas être pérennisée. L'évolution du nombre de livraison nécessite d'avoir une modification au processus de livraison d'un client. De plus, l'évolution de la demande va accroître les frais du personnel nécessaire pour maintenir une gestion pareille, D'où la nécessité d'intervention pour automatiser le service de livraison.

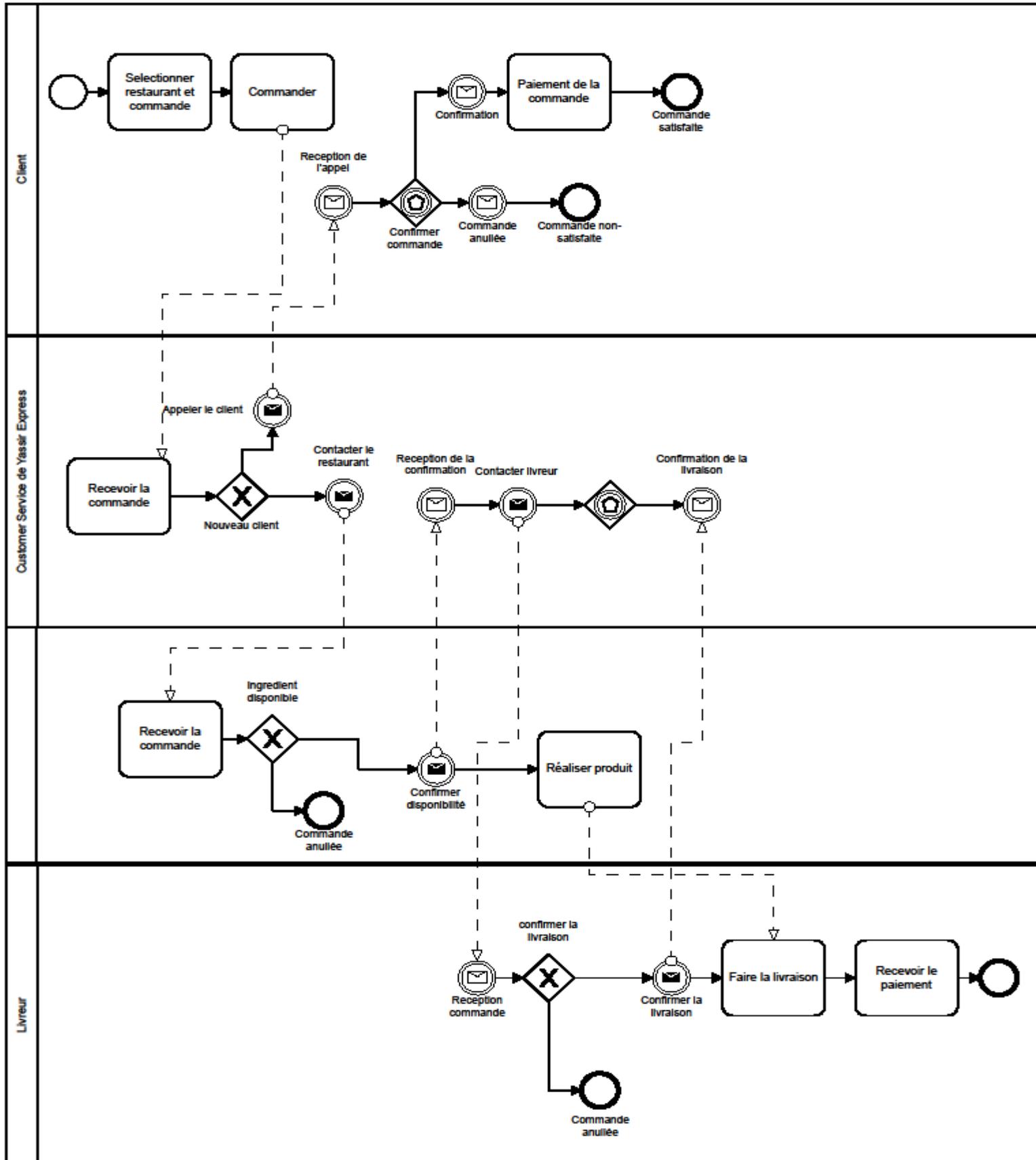


Figure 14 Diagramme BPMN global du service livraison de Yassir Express. L'intégralité des processus sont fait manuellement

1.10 Description de la problématique :

L'analyse stratégique et interne de Yassir Algérie permet d'évaluer la stratégie de Yassir globalement et les problématiques qui apparaissent dans le service de Yassir Express. Yassir Express est un seul exemple parmi les trois services cités où l'automatisation de la livraison est nécessaire.

L'automatisation de l'affectation des commandes au livreurs permettra d'améliorer la prestation de service. De plus, la solution à ce problème conduira à un meilleur bénéfice à l'entreprise, qui aura besoin de moins d'employer pour gérer les flux logistiques de la livraison.

En se basant sur ce dysfonctionnement, la problématique peut être formulée par la question suivante :

Comment optimiser l'affectation du service de livraison de Yassir Express tout en minimisant les coûts liés à l'activité ?

En premier lieu, une analyse des structures et processus existants dans l'entreprise a été réalisé pour comprendre le comportement des différents acteurs de l'entreprise. Un audit a été aussi réalisé pour déterminer les forces et les faiblesses du modèle existant.

En seconde partie, une revue scientifique sur l'ensemble des problèmes d'affectation de flottes logistiques est définie à l'aide de l'optimisation combinatoire. De plus, plusieurs approches de résolution sont mentionnées. Cette approche nous permettra de faire des hypothèses sur la problématique et introduire les solutions utilisées pour la dernière partie du rapport.

En dernière partie, une résolution de la problématique est abordée. Cette résolution suit une approche scientifique pour optimiser les flux de transport de l'entreprise.

2 Introduction au VRP et ses problèmes dérivés

2.1 Introduction :

Le problème des tournées de véhicule (*Vehicle Routing Problem*) est l'un des problèmes les plus étudiés dans la recherche opérationnelle (Toth and Vigo, 2014). Les variantes du problème VRP sont constamment étudiées et introduites, couvrant ainsi plusieurs domaines d'application en gestion de transport.

La première partie élabore une description détaillée du VRP. Le but est de se familiariser avec les différents concepts dont les fenêtres à temps, le ramassage et les problèmes dynamiques. En seconde partie, une revue du problème de tournées de véhicule avec ramassage est réalisée, le but est de donner une formulation au problème et d'expliquer sa relation avec les problèmes de VRP. En troisième partie, l'étude sera axée sur les différentes méthodes de résolution utilisées pour résoudre les problèmes d'optimisation combinatoire ; ces algorithmes sont classés en deux parties : des méthodes approximatives et des méthodes exactes. La troisième partie est fortement axée sur l'algorithme de colonies de fourmis (ACF)

Ce chapitre explique les notions de bases nécessaires pour comprendre la partie pratique de ce rapport.

2.2 Problèmes des tournées de véhicule :

Le problème du voyageur de commerce (TSP, *Traveling Salesman Problem*) est une extension au problème du VRP. Le problème de TSP répond à la question suivante « soit un ensemble de ville qu'un commerçant doit visiter un commerçant, sachant que la distance entre chaque ville est déterminée, qu'elle est le plus court chemin pour visiter l'ensemble des villes ? »

Le VRP est une généralité du TSP, le VRP permet de répondre à la question « étant donné un ensemble de véhicule de livraison, qu'elle est la configuration optimale pour livrer un ensemble de consommateur ? ». Donc, le problème de VRP consiste à trouver un circuit de distribution optimal pour chaque demande de client sous contraintes liées au véhicule, exigences clients et même parfois des contraintes environnementales.

Dans la prochaine section, on modélise le problème C-VRP (*Capacitated Vehicle Routing Problem*) Ce problème a fait l'objet de plusieurs études en milieu scientifique. Dans cette optique, le problème C-VRP est un problème VRP ou on suppose que les véhicules de transports ont une quantité de bien à transporter limitée. Le but de ce problème est de minimiser la distance parcourue par l'ensemble des véhicules de livraison, tout en livrant un bien unique à une liste déterminée de clients.

2.2.1 Formulation du problème :

A travers la littérature académique, il existe plusieurs formulations pour le CVRP. Chaque formulation a des avantages et inconvénients dans l'implémentation des solutions.

Cependant La formulation utilisée dans (Ralphs et al., 2003) permet d'expliquer facilement la problématique et ses contraintes.

On considère un ensemble de clients $i \in N = \{1, \dots, n\}$ qui ont une demande $d_i \geq 0$ d'un produit unique à partir d'un dépôt central $\{0\}$ en utilisant k véhicule de livraison où chaque véhicule à une capacité identique C . La livraison doit se faire à un coût minimal, ce coût peut être représenté de plusieurs manières, par exemple :

- La distance parcourue par les camions.
- Le temps de livraison pour les clients.
- Le moindre impact environnemental.

$c_{ij} \geq 0$ Représente le cout de déplacement entre de i a j tel que $0 \leq i, j \leq n$.

La solution à ce problème est représentée par un ensemble de route $\{R_1, \dots, R_k\}$ ou chaque route satisfait le critère $\sum_{\{j \in R_i\}} d_j \leq C$.

Ce problème est associé à un graphe complet consistant de $E = N \cup \{0\}$ sommets et des arêtes avec des couts $c_{ij}, \{i, j\} \in E$. La solution de ce graphe consiste à trouver k cycle ou l'intersection est le nœud dépôt.

Par conséquent, le problème devient un problème d'optimisation où $x_{i,j}$ est une variable de décision :

$$\begin{aligned} \min \sum_{e \in E} c_e x_e \\ \sum_{e=\{0,j\} \in E} x_e &= 2k \\ \sum_{e=\{i,j\} \in E, i \in S, j \notin S} x_e &= 2 \quad \forall i \in N \\ 0 \leq x_e \leq 1 \quad \forall e = \{i, j\} \in E, i, j \neq 0 \\ 0 \leq x_e \leq 2 \quad \forall e = \{0, j\} \in E \\ x_e \text{ integral } \forall e \in E \end{aligned}$$

- La fonction objective de cette modélisation minimise les coûts de transport des demandes de N clients.
- La première contrainte assure que chaque route commence et finit en dépôt.
- La deuxième contrainte assure que chaque client est visité par un seul véhicule.

- La troisième contrainte est une contrainte de capacité, $b(S) = (\sum_{i \in S} d_i) / C$, elle représente la borne inférieure du nombre de véhicule nécessaire pour satisfaire la demande des clients
- La quatrième, cinquième et sixième contrainte représentent la variable de décision

2.2.2 Complexité du problème :

La théorie de complexité est une branche des mathématiques qui vise à classifier les problèmes en algorithmique en termes de ressources nécessaires pour trouver une solution. Cette démarche permet de déterminer la difficulté à résoudre un problème indépendamment des machines utilisées pour le résoudre. Cette théorie utilise la notion d'une machine déterministe pour classifier l'ensemble des problèmes. Une machine déterministe est une machine théorique qui exécute un algorithme déterminé pour résoudre un problème.

Il existe principalement deux classes de problème en théorie de complexité :

- Classe P (*Polynomial Time*) : représente l'ensemble des problèmes dont il existe des algorithmes de résolution en temps polynomiale en utilisant une machine déterministe. En résumé la résolution de ces problèmes est facile.
- Classe NP (*Non-deterministic Polynomial time*) : Cette classe représente l'ensemble des problèmes dont on peut vérifier une solution en une durée polynomiale. Cependant, la résolution du problème est d'ordre exponentiel. Ces problèmes sont complexes. Il existe deux classes de problèmes NP :
 - NP-Complet : C'est l'ensemble des problèmes dont une vérification de la résolution peut s'effectuer en ordre polynomiale et dont tous les problèmes NP peuvent être réduits en problèmes NP-Complet à l'aide d'une réduction polynomiale.
 - NP-difficile : C'est l'ensemble des problèmes d'optimisation qui sont associés à un problème NP-Complet. L'un des exemples est le problème du voyageur de commerce.

Puisque le problème du VRP est une extension du TSP, le VRP est donc un problème NP-difficile.

2.3 Les variantes du VRP dans la littérature :

Le problème du VRP existe depuis plus de 50 ans. Depuis, de nombreuses variantes du VRP ont été citées. Cette section est une revue non-exhaustive des problèmes qui existent en littérature scientifique.

Dans cette partie, plusieurs caractéristiques des problèmes sont introduites. Ensuite, Les problèmes rencontrés en littérature sont introduits, ces problèmes sont souvent la combinaison des caractéristiques mentionnées précédemment.

2.3.1 Les fenêtres de temps :

Le Problème de tournées de véhicules avec contraintes et fenêtre de temps (*Vehicle Routing Problem With Time Windows*) regroupe l'ensemble des problèmes où la satisfaction des demandes des clients doit se faire dans un intervalle de temps déterminé (Toth and Vigo, 2014). Cette contrainte peut être définie comme une contrainte dure. Où dans ce cas la résolution n'a pas le droit de livrer un client sa demande en dehors de l'intervalle du temps défini par le client.

Par exemple, la livraison des journaux aux domiciles doit se faire en matinée seulement. Cependant, cette contrainte peut être définie à contrainte souple si le client peut être servi en dehors de l'intervalle prédéfini, mais avec une pénalité ; La livraison d'un produit sous cette contrainte est à contrainte dure s'il n'est pas possible de livrer un client dans un intervalle de jours connu, sinon le client est insatisfait.

2.3.2 Un seul ou plusieurs véhicules de transport :

Les problèmes du VRP peuvent avoir comme contrainte le nombre de véhicule utilisé pour la livraison. Les problèmes dit de tournée d'un seul véhicule « Single Vehicle Routing Problem » ont la caractéristique d'utiliser un seul véhicule de transport (Toth and Vigo, 2014). Un exemple réel étudié est le problème d'un seul véhicule de transport, appelé souvent le problème du voyageur de commerce.

Les VRPs avec plusieurs véhicules de transport peuvent aussi avoir plusieurs caractéristiques, par exemple les véhicules peuvent avoir des capacités de livraison différentes ou même avoir des véhicules qui n'ont pas accès à des clients. La configuration des véhicules de transport est divisée en deux parties :

- Véhicules homogènes : dans cette configuration, les véhicules de transport ont la même capacité de transport et caractéristique. D'une manière générale, n'importe quel véhicule choisi peut faire la tournée planifiée.
- Véhicules hétérogènes : Les véhicules de transport diffèrent en configuration. Ils ont des capacités de livraisons et caractéristiques différentes.

2.3.3 Un seul ou plusieurs dépôts :

Le VRP peut aussi avoir plusieurs dépôt (noté M-VRP). Dans ce cas, chaque dépôt a un nombre déterminé de véhicule pour satisfaire le besoin des clients. Une autre variation peut aussi exister dans ces problèmes, la possibilité de commencer à partir d'un dépôt et finir la tournée dans un autre.

Les dépôts peuvent aussi avoir des caractéristiques particulières, par exemple la surface de l'entrepôt peut accueillir un ensemble de véhicule déterminé ou même des fenêtres d'ouverture pour les entrepôts.

2.3.4 Tournée de véhicule avec ramassage :

Dans ce cas ,le véhicule de transport doit faire un ramassage dans un endroit et une livraison dans un autre. Le ramassage peut aussi être simultané, par exemple le problème de livraison de lait à domicile nécessite une livraison d'une bouteille pleine et le ramassage d'une autre vide.

2.3.5 Disponibilité de l'information :

Plusieurs paramètres qui rentrent dans la modélisation de la solution optimale peuvent être inconnu lors de l'élaboration du plan de transport. Dans ce cas, le VRP étudié est appelé un VRP dynamique(Toth and Vigo, 2014) .Un problème dynamique est un problème où on ne connaît pas une information importante pour pouvoir élaborer un modèle optimal. Les problèmes souvent étudiés sont généralement divisés en trois parties :

- Demande incertaine : Dans cette partie, le volume demandé par le client pour collecter ou livrer est incertain
- Client incertain : Dans ce cas, les clients peuvent apparaître lors de la réalisation d'une livraison.
- Temps incertain : le temps de disponibilité du client peut être incertain.

Par contre, un problème où l'intégralité de l'information nécessaire pour élaborer le modèle est disponible est appelé un problème statique.

2.3.6 Revue des problèmes VRP :

Dans cette partie, plusieurs problèmes variantes du VRP vont être introduits :

Tableau 5 Résumé de la nomenclature des problèmes dérivés du VRP

Nomenclature du problème	Description détaillé
Problème de tournées de véhicules avec fenêtres de temps (<i>Vehicle Routing Problem With Time Windows VRPTW</i>)	Dans cette configuration, des fenêtres de temps sont établies. Ces fenêtres à temps correspondent à l'heure où le client doit être livré.
Problème de tournées de véhicules avec ramassage (<i>Pickup and Delivery Problem</i>)	La demande des clients doit être ramassée d'un endroit, souvent désigné par le client, et livré dans un autre endroit déterminé.

Problème de tournées de véhicule avec demande dynamique (<i>Dynamic Vehicle Routing Problem</i>)	Si une partie nécessaire pour l'élaboration du plan de transport est indisponible, le VRP étudié est nommé VRP dynamique
---	--

Problème de tournées de véhicule général (<i>Generalized Vehicle Routing Problem</i> GVRP)	C'est la configuration la plus générale du VRP. Les demandes des clients ne sont pas homogènes dans ce cas. Certaines demandes peuvent avoir une fenêtre à temps, d'autres un critère de ramassage...en résumé, chaque commande client est unique
---	---

Souvent, les problèmes rencontrés en littérature sont une combinaison des caractéristiques décrites précédemment. Le tableau 6 suivant résume l'intégralité des problèmes mentionnés précédemment :

Tableau 6 Récapitulation des caractéristiques des VRP auxiliaires

Nom du problème	Fenêtre a temps	Nombre de véhicule	Nombre de dépôt	Ramassage	Nature de la demande	Capacité du véhicule
VRP	Libre	Multiple	Unique		Statique	Homogène
CVRP	Libre	Multiple	Unique		Statique	Homogène
VRPTW	Dures	Multiple	Unique		Statique	Homogène
PDP		Multiple	Unique	Nécessaire	Statique	Homogène
DVRP		Multiple	Unique		Dynamique	Homogène
GVRP		Multiple	Au moins un seul		Statique	Hétérogène

Les cases vides impliquent que le modèle peut avoir plusieurs configurations. Par exemple si la fenêtre de temps du problème PDP est dure, le problème se transforme en problèmes PDPTW (Problème de tournées de véhicules avec ramassage et fenêtre à temps, ou *Pickup and Delivery Problem With Time Windows*)

2.4 Problème de tournées de véhicule avec ramassage :

Le problème de tournées de véhicule avec ramassage est un cas particulier du VRP. Le problème a pour but de satisfaire des demandes de transports (bien ou client) d'une origine vers une destination. Ces types de problèmes sont classifiés selon les types de commandes et les structures étudié (Mitrović-Minić and Laporte, 2006). Il existe principalement trois sous-groupes :

- « Many-to-Many » : Cette classe de problèmes considère que chaque demande a plusieurs origines et destination et chaque location peut être l'origine ou la destination d'une commande. Parmi les exemples les plus connues, le problème de repositionnement des stocks entre plusieurs entrepôts
- « one-to-many-to-one » : Elle est caractérisée par deux types de demande, une demande de livraison vers une destination et une demande de collecte à partir de cette destination. Ces problèmes sont les plus présents dans la gestion des retours en logistique.
- « one-to-one » : La demande cette fois est caractérisée par deux points, une origine et une destination. Ces problèmes apparaissent souvent en services de livraison du courrier.

Dans la prochaine section, la formulation mathématique du PDP est réalisée, particulièrement le cas « problème de tournées avec ramassage et fenêtre à temps »

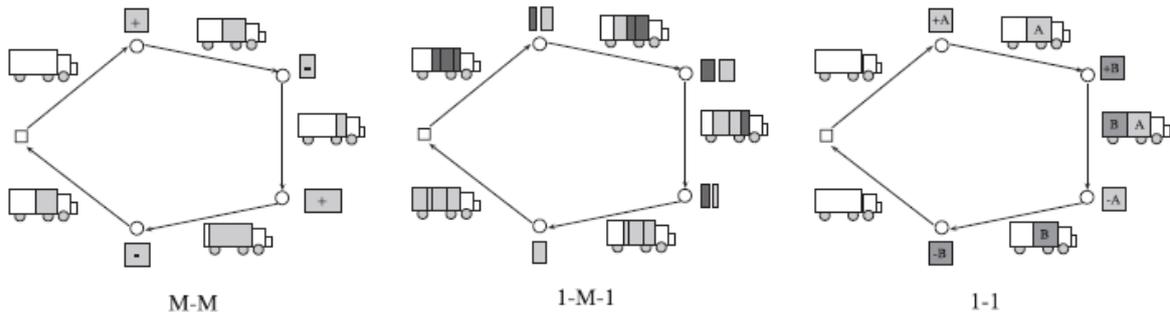


Figure 15 Les différents types de ramassage en PDP(Toth and Vigo, 2014)

2.4.1 Formulation du problème :

En suivant la formulation de (Cordeau, 2006), soit un graphe $G = (V, A)$ ou V est l'ensemble des sommets $V = P \cup D \cup \{0, n + 1\}$ ou $P = \{1, \dots, n\}$ représente l'ensemble des nœuds de ramassage et $D = \{n + 1, \dots, 2n\}$ l'ensemble des nœuds de livraison.

- q_i ou $i \in P$ représente la quantité à ramasser d'un endroit i pour livrer $n + i \in D$
- c_{ij} représente la distance pour faire se déplacer d'un point i vers un point j .
- s_i ou $i \in V$ représente le temps de service pour réaliser la tâche nécessaire (le ramassage et la livraison nécessite un temps pour la réalisation)
 - On suppose que $s_0 = s_{2n+1} = 0$, $q_{n+i} = -q_i$ et $q_0 = q_{(2n+1)} = 0$
- $[a_i, b_i]$ ou $i \in V$ représente les fenêtres à temps associé à un nœud, a_i et b_i représente le plus tôt et le plus tard fenêtres de service, respectivement.
- chaque arrete $(i, j) \in A$ est associé à un coût $c_{ij} \geq 0$ et un temps de voyage $t_{ij} \geq 0$
- La variable binaire x_{ijk} est égal a 1 si un véhicule $k \in K$ voyage de i a j , sinon $x_{ijk} = 0$
- T_{ik} représente le temps du début de service de k au nœud i

A l'aide cette formulation on obtient le problème d'optimisation suivant :

$$\begin{aligned}
 \text{Min} \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ijk} & \quad 1 \\
 \sum_{k \in K} \sum_{j \in V} x_{ijk} = 1 \quad \forall i \in P & \quad 2 \\
 \sum_{j \in V} x_{ijk} - \sum_{j \in V} x_{n+i,jk} = 0 \quad \forall k \in K, i \in P & \quad 3
 \end{aligned}$$

$$\begin{aligned} \sum_{j \in V} x_{0jk} &= 1 \quad \forall k \in K & 4 \\ \sum_{j \in V} x_{jik} - \sum_{j \in V} x_{ijk} &= 0 \quad \forall k \in K, i \in P \cup D & 5 \\ \sum_{j \in V} x_{i,2n+1,k} &= 1 \quad \forall k \in K & 6 \\ T_{jk} &\geq (T_{ik} + s_i + t_{ij})x_{ijk} \quad \forall k \in K, i \in V, j \in V & 7 \\ Q_{jk} &\geq (Q_{ik} + q_i)x_{ijk} \quad \forall k \in K, i \in V, j \in N & 8 \\ T_{n+i,k} - T_{ik} - s_i - t_{i,n+i} &\geq 0 \quad \forall i \in P & 9 \\ a_i &\leq T_{ik} \leq b_i \quad \forall i \in V, k \in K & 10 \\ \max\{0, q_i\} &\leq Q_{ik} \leq \min\{Q_k, Q_k + q_i\} \quad i \in V & 11 \\ x_{ijk} &\in \{0, 1\} \quad \forall i \in V, j \in V, k \in K & 12 \end{aligned}$$

- La première équation minimise le coût de transport
- 2 et 3 assure que chaque commande est satisfaite une seule fois et que l'endroit de ramassage et livraison sont visités par un seul véhicule.
- 4 ,5 et 6 assure que chaque véhicule commence sur le dépôt et finit dans un dépôt de destination
- Les contraintes 7 et 8 considèrent le respect des contraintes de temps et quantité à livrer, respectivement
- Contrainte 9 assure que les nœuds de ramassage est visité avant le nœud de livraison
- Contrainte 10 et 11 impose les contraintes de quantité qui peut transporter le véhicule ainsi que l'heure de visite du nœud, respectivement.

En résumé, l'algorithme reçoit un ensemble de commande et une flotte de véhicule. Chaque commande est associée a un endroit de ramassage et de livraison , et chaque véhicule a une quantité limitée. Le but est de trouver la meilleure affectation des commandes aux véhicules tout en minimisant la distance globale parcourue par les livreurs.

2.5 Résolutions des VRPs :

Puisque la complexité des problèmes VRP est d'ordre exponentiel. Trouver des solutions exactes aux problèmes VRP reste souvent une tâche difficile. Cependant il existe des méthodes de résolution optimales pour des problèmes de taille petites (jusqu'à 50 clients) et les problèmes les plus complexes sont souvent résolu avec des heuristiques(Toth and Vigo, 2014).

2.6 Méthodes de résolution exactes :

Enumérer l'intégralité des solutions possibles n'est pas possible même pour les petits problèmes. Cependant, il existe plusieurs approches en programmation en nombre entier, ces

approches permettent la résolution de problème plus complexe sans énumérer l'intégralité des problèmes. Ces méthodes sont une amélioration par rapport à l'énumération mais la résolution du problème reste aussi complexe.

Dans cette section, trois méthodes de résolution exactes sont introduites, il s'agit du Branch and Cut, Branch and Bound, et la méthode la plus récente le Branch and Cut and Price. Cette dernière est une approche récente pour résoudre exactement les VRP, et cette approche donne les meilleurs résultats parmi les trois méthodes(Toth and Vigo, 2014).

En pratique, même l'approche de ces algorithmes peut ne pas être exacte. Il existe même des instances où ces algorithmes n'ont pas pu résoudre des problèmes à 50 éléments exactement(Toth and Vigo, 2014). Ceci laisse un large champ de recherche pour trouver de meilleures approches.

2.7 Branch and Bound :

Branch and Bound ne représente pas seulement une technique pour résoudre les problèmes d'OLNE (optimisation linéaire en nombre entier) mais une approche utilisée pour plusieurs types de problèmes(Casquilho, n.d.). Cette approche est utilisée par plusieurs Solver existant (IBM CPLEX, Gurobi ...)

Branch and Bound part du principe que la solution globale d'un problème OLNE est la résolution optimale de plusieurs sous-problèmes, ces sous-problèmes sont en relation avec le problème principal. En gardant un historique des meilleures solutions existantes sur ces sous-problèmes, la meilleure solution du problème générale est la meilleure solution trouvée en un sous-problèmes particulier.

Comment on trouve ces sous-problèmes ? Branch and Bound utilise le principe de la relaxation. La relaxation transforme les variables de décision d'une variable entière (ou binaire) en variable réelle.

Pour pouvoir mieux comprendre le principe de Branch and Bound, un exemple tiré de (Casquilho, n.d.) nous montre l'approche globale de Branch and Bound.

Soit le problème d'optimisation en nombre entier suivant :

$$\text{maximiser } Z = 100x_1 + 150 x_2$$

$$\text{s. c } 8000x_1 + 4000x_2 \leq 40000$$

$$15x_1 + 30x_2 \leq 200$$

$$x_1, x_2 \geq 0$$

Dans un premier degré, si on suppose que x_1, x_2 sont des variables réels la solution optimale du problème est simple. Dans notre cas :

$$x_1 = 2.22 \quad x_2 = 5.56 \quad Z = 1055,56$$

Cette solution optimale du problème est appelée borne supérieur (Upper Bound ou UB), c'est la solution optimale pour le problème en nombre réels. Si on arrondit les variables x_1, x_2 la solution obtenue est :

$$x_1 = 2 \quad x_2 = 5 \quad Z = 950$$

Cette solution qui satisfait les critères du problème OLNE est appelée borne inférieur (lower Bound ou LB). Si aucune autre solution qui satisfait les critères OLNE existent, cette solution est la solution optimale au problème.

Pour trouver d'autre solution possible, l'approche demande de rajouter d'autre critère au problème et de tester la faisabilité de cette solution dans le problème OLNE. Par exemple, pour trouver d'autres problèmes dans notre cas, on rajoute deux problèmes avec les conditions suivantes :

Pour le premier probleme : maximiser $Z = 100x_1 + 150 x_2$

$$s. c \quad 8000x_1 + 4000x_2 \leq 40000$$

$$15x_1 + 30x_2 \leq 200$$

$$x_1 \geq 0$$

$$x_2 \geq 5$$

Pour le second probleme : maximiser $Z = 100x_1 + 150 x_2$

$$s. c \quad 8000x_1 + 4000x_2 \leq 40000$$

$$15x_1 + 30x_2 \leq 200$$

$$x_1 \geq 0$$

$$x_2 \leq 5$$

Le but de cette approche est de trouver d'autre solution avec des nombres entiers. Le diagramme suivant résume l'intégralité des branches réalisé pour trouver la solution optimale pour l'OLNE qui se trouve en le sixième nœud.

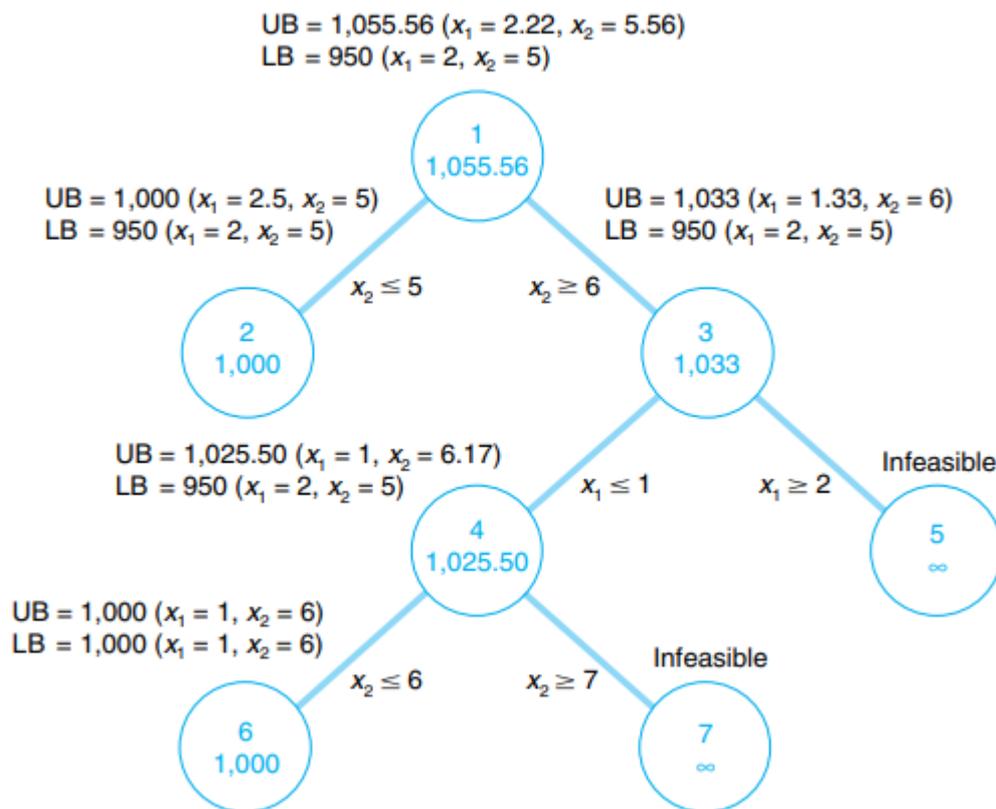


Figure 16 Diagramme qui résume l'approche Branch and Bound appliqué au problème considéré(Casquilho, n.d.)

La méthode Branch and Bound a plusieurs avantages et inconvénient. Cette méthode est considérée comme l'approche la plus fiable pour trouver une solution exacte. Cependant, la complexité de cette approche rend la résolution des problèmes plus complexe au VRP impossible à résoudre, même avec les ordinateurs les plus puissants existants en entreprise.

2.8 Branch and Cut:

Cette méthode est une amélioration de l'approche précédente Elle utilise la même approche que Branch and Bound, mais au lieu de rajouter que des critères liés au variable, on rajoute des contraintes pour améliorer la rapidité des solutions trouvés. Ces contraintes rajoutées au problème principal proviennent d'une méthode appelée méthodes des plans sécants (*Cutting Planes Method*)(Cordeau, 2006).

Il existe plusieurs méthodes pour ajouter des contraintes(Marchand and Wolsey, 2001), on peut citer :

- La méthode de Gomory
- Les inégalités MIR (*Mixed Integer Rounding*)

L'avantage principal de cette approche est une possibilité de trouver la solution optimale plus rapidement que l'approche Branch and Bound. De plus, même si l'approche reste complexe, elle est considérée comme une amélioration par rapport à l'approche Branch and Bound en termes de complexité.

2.9 Branch and Cut and Price:

Branch and Cut and Price est la méthode la plus générale pour résoudre les problèmes OLNE. Elle généralise le principe des branches trouvé en Branch and Bound et le principe des coupes en Branch and Cut en rajoutant la méthode de génération de colonne.

La génération de colonne part du principe que la majorité des variables utilisées dans le programme linéaire sont nulles (Toth and Vigo, 2014). De ce fait, il n'est pas nécessaire d'utiliser l'intégralité des variables mais de commencer l'optimisation par des variables définies et ensuite générer des colonnes (variables) dans notre modèle global.

Il existe de nombreuses approches pour générer les colonnes en pratique. Ces approches dépendent souvent du problème étudié. Puisque les variables de décision du VRP sont souvent nulles, l'approche Cut and Price est considérée, jusqu'à aujourd'hui, comme la meilleure méthode exacte pour résoudre les problèmes VRP.

Cette approche a une meilleure convergence avec une complexité réduite. Cependant, l'implémentation de ces algorithmes est souvent difficile, et dépendant du problème considéré.

2.10 Conclusion :

Malgré le développement récent des méthodes de résolution des OLNE, les méthodes exactes ont une utilisation pratique limitée pour résoudre les VRP. (Toth and Vigo, 2014) montre qu'il existe des instances de VRP non résolues à l'optimum avec seulement 50 éléments (dépôts, véhicules et clients) considérés.

2.11 Méthodes de résolution approximatives :

Les méthodes de résolutions approximatives pour résoudre les problèmes combinatoires. Le but de cette approche est de générer une solution en trouvant un équilibre entre l'efficacité pratique de cette solution et la complexité de la méthode de résolution.

Les méthodes décrites prochainement sont souvent des approches utilisées pour résoudre les VRP. En pratique, plusieurs approches sont combinées pour trouver des solutions optimales. La combinaison des algorithmes est souvent appelée des algorithmes hybrides.

2.12 Approches Métaheuristiques :

Les métaheuristiques sont des approches utilisées pour trouver des solutions approximatives en un temps faible. Souvent, les solutions retournées par les métaheuristiques ne sont pas optimales, mais cette solution est souvent peu variable par rapport à la solution optimale. La perte liée à l'optimalité est souvent gagnée en performance et robustesse de la solution (Luke, 2013). De ce fait, ces approches heuristiques sont souvent utilisées en pratiques pour résoudre le problème du VRP et TSP avec d'excellente approximation à la solution optimale.

Il existe plusieurs classifications pour les méthodes métaheuristiques. Cependant, la classification la plus globale est la suivante (Abd-Alsabour, 2016) :

- Heuristiques inspirés par la nature : ces algorithmes sont influencés par les processus biologique existant en nature, par exemple la mutation en biologie ou le recuit en physique. Parmi les algorithmes existants on peut citer :
 1. Algorithme Génétique
 2. Programmation génétique
 3. Algorithme des colonies de fourmis
- Heuristiques non-inspiré par la nature : Ces algorithmes ont été développés par plusieurs chercheurs sans une inspiration générale. Parmi ces algorithmes on peut citer :
 1. La recherche locale
 2. Recherche Taboue.
 3. Heuristique glouton

Parmi les heuristiques existantes. Plusieurs vont être présentés dans cette section. Cette revue ne couvre pas l'intégralité des approches heuristique utilisée pour résoudre les problèmes VRP.

2.13 Recherche locale :

Le but de la recherche locale est de trouver une solution optimale à partir d'une solution initiale (Luke, 2013). Les étapes de l'approche sont :

- Première étape : Trouver une solution initiale X qui satisfait les critères du problème étudié, par exemple pour un problème C-VRP, une solution initiale réalisée satisfait les critères de capacité et du VRP est initié.
- Deuxième étape : En utilisant la solution initiale, une légère modification de cette solution est réalisée. Cette modification dépend souvent des problèmes étudiés et de l'approche choisit par l'utilisateur. Par exemple, pour le même problème CVRP expliqué précédemment, la modification peut consister d'un changement de

l'affectation d'une commande d'un véhicule à un autre tout en satisfaisant les critères de CVRP. Cette solution générée est enregistrée pour la prochaine étape

- Troisième étape : Dans cette partie, On compare la solution initiale et la solution créée. Si la solution créée à un meilleur résultat, cette solution est gardée pour de prochaine itération du problème, sinon la solution initiale est gardée. Par exemple, dans le cas du problème CVRP, la fonction objective tranche entre la qualité des deux solutions, et la meilleure est gardée.
- Quatrième étape : cette approche est répétée plusieurs fois, pour générer la meilleure solution.

Cette méthode permet généralement de trouver un optimum local pour un ensemble de solution. Cependant cet optimum est non optimal par rapport à l'optimum global de l'espace des solutions possibles. De ce fait, cet algorithme est appelé un algorithme purement d'exploitation car il exploite une solution initiale pour trouver une solution optimale.

L'inconvénient principal de cet algorithme est la possibilité d'être coincé dans une solution non optimale. Pour y remédier à ce problème plusieurs approches peuvent être utilisées :

- Élargir le champ de modification de la solution initiale : une modification plus globale de la solution initiale permet de sortir de l'optimum local.
- Réinitialiser la solution initiale : Après avoir exploité une solution initiale pour trouver son optimum local, on génère une autre solution initiale aléatoire et on exploite cette solution. A la fin, La solution qui génère le meilleur résultat est gardée.

Cette méthode représente un élément de base pour les approches qui vont être introduite prochainement, car elle permet d'avoir des résultats initiaux à améliorer.

2.14 Recherche Tabou :

Cette méthode est souvent considéré comme une généralisation de la recherche locale (Luke, 2013). Elle part du principe que la recherche locale peut souvent tomber sur un optimum local ; pour y remédier à ce problème, l'algorithme garde un ensemble de solution retourné par un algorithme d'exploitation (comme la recherche locale) pour ne pas être utilisé prochainement, car ces solutions sont déjà exploitées.

Pour mieux illustrer l'algorithme, la figure suivante illustre l'algorithme tabou.

- Première étape : Initialiser une solution au problème considéré, cette initialisation doit se faire rapidement, car le plus important dans cet algorithme est l'optimisation des solutions.
- Deuxième étape : Créer une liste de solution à l'aide d'un algorithme de recherche locale. A la fin de cette étape la meilleure solution est retournée par l'algorithme.

- Troisième étape : Si la solution a un meilleur résultat par rapport à d'autre candidat dans une liste taboue, cette solution est enregistrée dans la liste. La liste taboue contient souvent une limite sur le nombre de solutions qui peuvent être enregistré sur elle ; de ce fait, si la liste taboue atteint cette limite, les solutions les plus anciens sont supprimées.
- Quatrième étape : si le critère d'arrêt de l'algorithme est satisfait, l'algorithme retourne la meilleure solution trouvée.

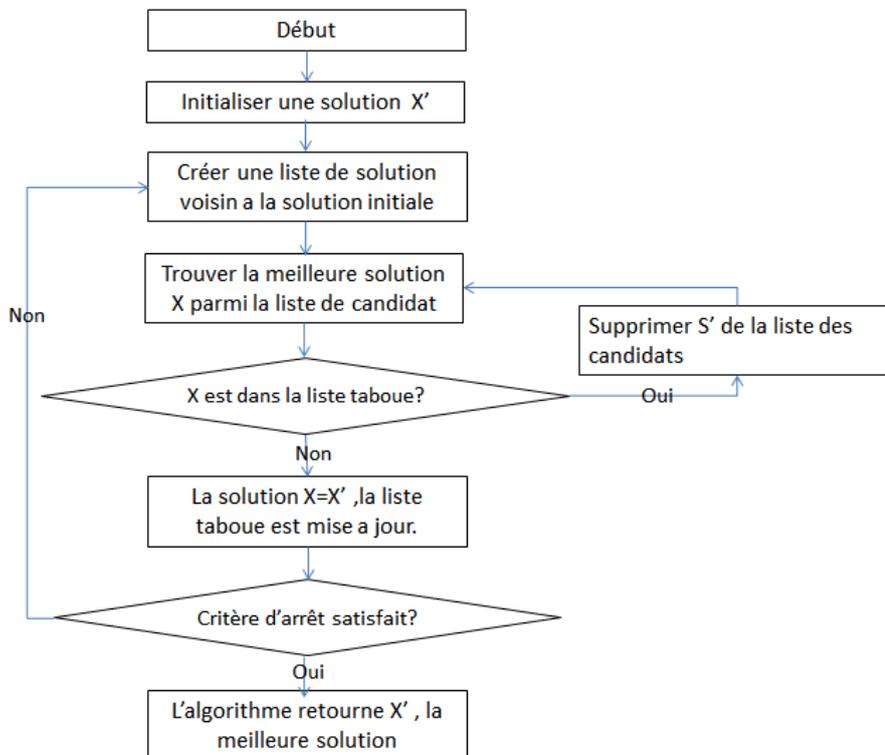


Figure 17 Résumé de l'approche recherche tabou

2.15 Recuit simulé :

La méthode du recuit simulé a été inspirée par le processus de refroidissement des métaux. L'alternance des cycles de refroidissement pour les métaux augmente la dureté de ce dernier. Souvent, le temps de refroidissement des métaux est réduit au fur et à mesure pour avoir à la fin un métal très solide.

Comme ce processus, en métaheuristique on introduit plusieurs paramètres fictifs qui sont :

- L'énergie E : elle représente la fonction objective du problème étudié.
- La température T : le principe de la température est abstrait, il s'agit d'une fonction décroissante. Le but de la température est de donner un espace large d'exploration

des solutions. Ensuite, après chaque itération et selon une probabilité, l'espace est réduit pour permettre une meilleure exploitation de la solution trouvée.

Au début, une solution initiale est choisie aléatoirement parmi l'espace des solutions considéré par le problème. Cet état correspond à une énergie E_0 . Une température T_0 est aussi choisi pour permettre une exploration large des solutions possible.

Pour chaque itération de l'algorithme, une modification de la solution initiale est réalisée. Si cette variation optimise la fonction objective (baisse l'énergie E_0), elle est ensuite utilisée pour d'autre itération de l'algorithme.

A la fin de cette étape, deux critères montrent si une solution est gardée ou pas. Si la solution générée, supposons son énergie est E_1 , représente une amélioration par rapport à la solution initiale E_0 , cette nouvelle solution est donc gardée. Cependant, même si la solution E_1 n'est pas meilleure par rapport à la solution trouvée, $\exp\left(\frac{(E_0 - E_1)}{T}\right)$ représente la probabilité de garder E_1 pour les prochaines itérations. Le but de cette méthode est d'essayer, malgré la non-optimalité d'une solution E_1 , de trouver d'autre piste à explorer dans l'espace des solutions.

Après chaque itération de l'algorithme, la température T est souvent réduite. Cette réduction permet à l'algorithme d'explorer de moins en moins de nouvelles pistes. En parallèle, une meilleure exploitation des optimums locaux déjà établi.

Durant toutes ces itérations, l'algorithme enregistre la solution optimale. Cette solution va être rendue par l'algorithme, une fois le critère d'arrêt est satisfait. Le diagramme suivant résume l'approche établi du recuit simulé.

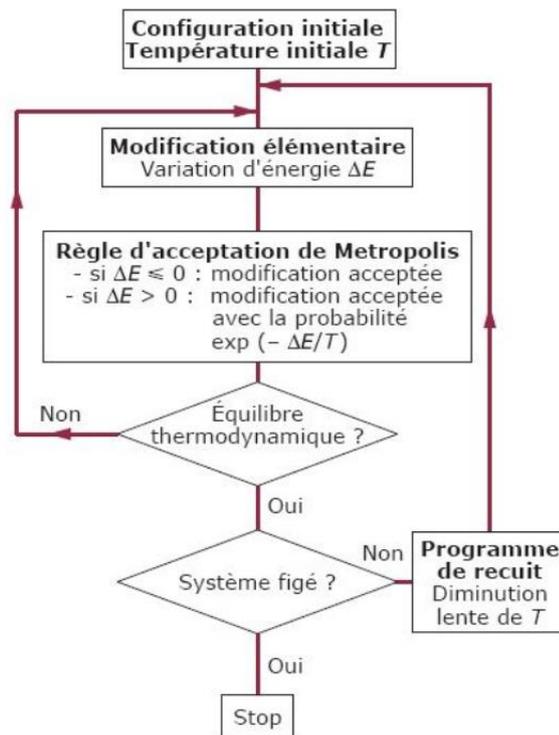


Figure 18 Résumé du recuit simulé

2.16 Algorithmes colonies de fourmis (ACF) :

L'algorithme de colonies de fourmis est parmi les algorithmes les plus utilisés pour résoudre les problèmes VRP et TSP. L'avantage principal de cet algorithme est la non nécessité d'utiliser une solution initiale pour trouver une solution. Ces types d'algorithmes sont appelés algorithmes constructeurs, car l'initialisation de l'algorithme ne nécessite pas une présence d'une solution initiale, mais l'algorithme construit sa propre solution et l'améliore durant plusieurs itérations (Luke, 2013).

2.17 Inspiration :

L'algorithme s'inspire du phénomène d'approvisionnement (*foraging*) des fourmis. Durant le XX^e-ème siècle, Pierre-Paul Grassé a observé que le comportement des termites est affecté par un stimulateur pour les insectes d'une même colonie ; ce stimulateur aide les autres termites à trouver un chemin optimal vers une nourriture, ce chemin contient la plus grande densité de ce stimulateur, a-t-il observé (Dorigo et al., 2006).

L'un des exemples les plus connus est observé dans les colonies de fourmis. Les fourmis déposent une substance, appelée phéromone, qui aide les fourmis à trouver le chemin optimal vers une concentration d'aliment. **Si une fourmi se trouve dans une intersection de deux chemins, le chemin contenant plus de phéromone a une probabilité plus élevée d'être choisi.**

Pour illustrer le rôle des phéromones, la figure ci-dessous présente un pont élaboré entre une colonie de fourmis (F) et une concentration de nourriture (N). Au début, à cause de l'inexistence de phéromone sur la route, le choix du chemin d'une fourmi est aléatoire, c'est la partie d'exploration des routes possibles. Cependant, à cause de la phéromone et de l'évaporation, le plus court chemin arrive à garder une concentration supérieure de phéromone par rapport au plus long chemin ; ceci mène plus de fourmis à prendre le chemin le plus court. Enfin, un chemin avec une forte densité en phéromone est le chemin le plus choisi par les fourmis, c'est le chemin optimal.

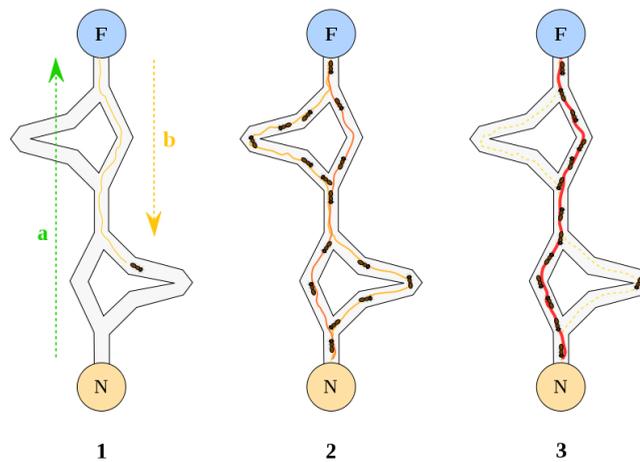


Figure 19 L'expérience des doubles ponts des colonies de fourmis

En pratique, cette approche aide les fourmis à trouver le chemin optimal dans un temps record et de déplacer la nourriture vers le nid d'une manière très ordonnée.

2.18 Métaheuristique :

En métaheuristique, l'algorithme des colonies de fourmis est souvent utilisé pour résoudre le problème du sac à dos ou du TSP (Dorigo et al., 2006).

L'algorithme ACF a principalement cinq-étapes :

- Construction des solutions candidatent
 - Le nombre de solutions est souvent expliqué par le nombre de fourmis de la solution.
- Recherche locale d'une ou plusieurs solutions (optionnelles)
- Mesurer la qualité de la solution finale choisit
- Mettre un processus d'évaporation de phéromone dans les routes choisies
- Améliorer le degré des phéromones des routes sélectionnées dans la solution optimale.

La résolution du problème de voyageur de commerce illustre mieux l'ACF.

2.18.1 Voyageur de commerce : algorithme de colonies de fourmis :

On considère un problème de voyageur de commerce où :

- V représente l'ensemble des villes que le voyageur de commerce doit visiter
- L'ensemble $E = \{(i, j) | i, j \in V\}$ représente l'ensemble des routes qui relie une ville i à une autre j , on considère que le voyageur peut se déplacer vers tous les autres nœuds à partir d'un nœud $i \in V$
- $c_{i,j} \geq 0 \forall i, j \in V$ représente la distance entre une ville i et une autre j . De ce fait, $c_{ii} = 0 \forall i \in V$ Ces distances sont souvent représentées par une matrice carrée de distance C .

Le but du voyageur de commerce est de faire une tournée sur chaque ville dans l'ensemble V en les visitant une et une seule fois. En théorie des graphes, il s'agit de trouver un chemin où on passe une et une seule fois par un sommet dans un graphe $G = (V, E)$, c'est le principe du chemin Hamiltonien.

$L(r) = \sum_{i \in r} d_{r_i r_{i+1}}$ Représente la distance parcourue par le voyageur de commerce dans une route r qui contient l'ensemble des villes de V . Le but principale de l'algorithme est de minimiser cette distance.

Le problème de voyageur de commerce est associé à un système, appelé système de fourmis (Ant System) :

- $b_i(t)$ ou $i \in V$ est le nombre de fourmis dans la ville i à l'instant t
- $m = \sum_{i \in V} b_i$ est le nombre total de fourmis
- $\tau_{i,j}$ ou $i, j \in V$ représente la quantité de phéromone présente dans un chemin (i, j) , T représente la matrice des phéromones. Les phéromones permettent à l'algorithme de mettre en exergue le degré d'utilisation d'une route. Plus un chemin est utilisé dans une solution optimale, plus ce chemin aura de phéromone.
- $\eta_{ij} = \frac{1}{d_{ij}}$ représente la visibilité d'une ville j si on est placé sur une ville i .

En pratique, chaque fourmi va commencer dans une ville aléatoire et faire son circuit Hamiltonien sur l'ensemble des villes. Ces fourmis artificielles ont la capacité de mémoriser les villes déjà visité, ainsi les tours Hamiltoniennes sont réalisables.

2.18.1.1 Choix des villes :

Pour chaque fourmis k sur une ville i , le choix d'une ville non visité j se fait selon la visibilité d'une ville et la quantité de phéromone déposée sur le chemin entre les deux villes i et j . En pratique, le choix est fait d'une manière aléatoire en utilisant la loi de densité de probabilité suivante :

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha * \eta_{ij}^\beta}{\sum_{j \in N_i^k} \tau_{ij}^\alpha * \eta_{ij}^\beta}, j \in N_i^k \\ 0 \text{ sinon} \end{cases}$$

N_i^k Représente l'ensemble de villes pas encore visité par la fourmi k. α et β Représente des paramètres d'ajustement. Ces paramètres contrôlent l'importance relative du taux de phéromone, ou de visibilité, sur le choix de la prochaine destination.

2.18.1.2 Mise à jour des phéromones :

Après la constitution de chaque route, la matrice des phéromones est mise à jour selon la formule :

$$\tau_{ij} = \rho \cdot \tau_{ij} + \Delta\tau_{ij} \text{ ou } i, j \in V \quad (1)$$

Cette approche modifie la phéromone de chaque route utilisée par les fourmis. En effet, ρ est appelé le taux d'évaporation de phéromone durant la constitution du cycle hamiltonien. $\Delta\tau_{ij}$ Représente la quantité de phéromone déposée par les fourmis durant la constitution de leur propre cycle. Ainsi, un chemin choisi par les fourmis bénéficiera d'une augmentation de la quantité de phéromone dans un chemin (i, j) , et donc plus de chance d'être choisi durant les prochaines itérations (le choix des villes est décrit précédemment).

En pratique, trouver une valeur de $0 \leq \rho \leq 1$ qui équilibre l'évaporation de phéromone est une tâche difficile. Une valeur ρ proche de 0 rend la solution dépendante seulement sur la visibilité des villes. Par contre une valeur proche de 1 rend l'algorithme dépendant de solutions déjà établies.

2.18.1.3 Quantité de phéromone déposée :

On appelle un cycle $T_k = (u_1, \dots, u_n)$ le cycle réalisé par la fourmi k, L_k représente la longueur de ce cycle. La quantité de phéromone déposée sur le chemin (i, j) par une fourmi k est défini par :

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} \text{ si } (u \in T_k, u = (i, j)) \\ 0 \text{ sinon} \end{cases} \quad (2)$$

Q est une constante. On remarque qu'une quantité plus élevée de phéromone sera placée sur les cycles avec une faible distance parcourue, car on considère les chemins pris dans ce cycle comme les routes constituant les plus probablement présentes dans la route optimale. De ce fait, la quantité de phéromone globalement déposée est ainsi :

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k$$

Où m représente le nombre de fourmi dans l'algorithme

2.18.2 Déroutement de l'algorithme :

Les étapes suivantes représentent l'évolution de la résolution du problème TSP avec l'approche ACF :

- 1) Les fourmis sont réparties aléatoirement sur les n villes
- 2) Les dépositions de phéromone sont initialisées à une valeur égale à c
- 3) Chaque fourmi choisie la liste de ville qu'elle visite, selon le critère de choix des villes.
- 4) Après n itération, l'intégralité des fourmis ont créé leur tour, chaque tour T^k a une longueur L_k
- 5) Les dépositions de phéromone sur chaque route utilisée par l'algorithme sont mises à jour à l'aide des formules (1) et (2)
- 6) Parmi les routes élaborées par les fourmis, on enregistre la route avec la longueur minimale si cette route est la meilleure tour jusqu'à maintenant.
- 7) Répéter à partir de l'étape 1 tant qu'un critère d'arrêt n'est pas satisfait. Souvent, le critère d'arrêt est le nombre d'itération de cet algorithme, établi à 50 ou 100 fois. Le nombre d'itération est noté a

2.18.3 Initiation des constantes :

$\alpha, \beta, \rho, Q, m$ et a Jouent un rôle important dans la rapidité de convergence vers la solution optimale. Cependant, il existe une réelle difficulté d'établir des valeurs qui facilitent la convergence, car aucune combinaison de paramètre ne peut fonctionner pour tous les types de TSP.

En pratique, les paramètres les plus utilisés sont les suivants :

Tableau 7 Les valeurs suggéré pour l'ACF

Paramètres	Valeurs
m	n
ρ	0,5
β	5
α	1
Q	100
c	<i>petite valeur positive</i>

2.18.4 Avantages et inconvénient de l'ACF :

L'ACF possède plusieurs avantages par rapport aux algorithmes décrits précédemment, on peut citer :

- Facilité à résoudre les problèmes d'optimisation d'ordre statique ou dynamique.
- L'approche constructive de l'algorithme donne la possibilité de résoudre les problèmes dérivés du TSP et VRP facilement par rapport aux autres approches métaheuristiques (Algorithme génétique, Optimisation par essais particulières)
- La possibilité de combiner cet algorithme, par exemple la recherche locale, pour améliorer encore plus les solutions proposées.

Par contre, l'inconvénient principal est la nécessité de choisir des bons paramètres pour avoir une convergence rapide et d'éviter les optimums locaux. De plus, l'ACF est considéré comme un algorithme qui a plus de complexité en implémentation par rapport à d'autre algorithme métaheuristiques.

2.19 Conclusion :

Dans ce second chapitre, plusieurs concepts nécessaires pour la relation du projet ont été abordés. Ces points sont déclinés principalement sur trois axes :

- Le premier axe définit les problèmes VRP rencontré en littérature scientifique ainsi que les modèles de programmation combinatoire utilisé pour résoudre le problème
- Le deuxième axe explique une classe particulière des problèmes VRP, à savoir les problèmes de livraison avec ramassage PDP, cette classe est le sujet principal de l'intervention au niveau de l'entreprise.
- Le troisième axe résume plusieurs méthodes de résolution de problèmes d'optimisation combinatoire. L'étude est fortement axée sur l'algorithme de colonies de fourmis, car elle est l'approche principale utilisée pour réaliser l'intervention en entreprise.

Après la présentation de l'état des lieux et de l'art, le troisième chapitre introduit une méthodologie pour résoudre la problématique introduite au premier chapitre.

3 Élaboration du modèle et solution proposée

3.1 Introduction :

Après avoir analysé la littérature scientifique, ce chapitre tente de résoudre une partie importante de la problématique à travers la réalisation d'un modèle mathématique et la définition d'une approche de résolution.

Au début, le problème rencontré en entreprise est modélisé et les contraintes exigées pour développer le projet sont définies. Des approches de résolution rencontrées en littérature scientifique sont aussi introduites. Ensuite, le choix d'une approche parmi les autres est justifié et une description en détail de l'approche est réalisée. A la fin, une partie importante de l'approche démontrée est résolue, à savoir la programmation d'un système de résolution du problème de voyageur de commerce avec ramassage et livraison. La dernière partie est consacrée à montrer les avantages et limites du modèle élaboré.

3.2 Classification et caractéristiques du problème :

A l'aide des deux chapitres précédents, une classification de la problématique peut être établie, cette démarche aide à définir ainsi une approche pour résoudre le problème.

Le cas étudié en entreprise est un problème dynamique de ramassage et livraison avec fenêtre à temps. Le problème est dynamique pour plusieurs raisons, particulièrement :

1. La demande de commande est incertaine : un client peut faire une commande de produit à tout moment. Cependant, la majorité des commandes de livraison sont faites entre 11h à 13h et entre 19h et 21h.
2. Les horaires de travail des livreurs est incertaine : un livreur a une grande liberté sur le choix des heures de travail. Il a aussi la possibilité de ne pas accepter l'offre de livraison, sans être pénalisé.
3. Les horaires d'ouverture des restaurants sont dynamiques : parfois des commandes peuvent être retardées à cause d'une ouverture tardive du restaurant choisi par le client.
4. Le temps de préparation de commande change : une commande au même restaurant peut prendre moins hors les heures de pointes, mais plus dans d'autres cas.

Les commandes de client peuvent avoir des fenêtres à temps souples ou dures. Cependant, opter pour un modèle sans fenêtres à temps reste possible.

L'objectif de l'automatisation de la livraison est de minimiser la distance parcourue par le livreur. Cette contrainte a été exigée par l'entreprise car :

5. Les livreurs globalement souhaitent faire de court trajet, même si, en apparence, ils sont moins payés.
6. En pratique, le bénéfice généré par un livreur est plus élevé lorsque le livreur parcourt une petite distance, car le bénéfice de livraison est composé d'un forfait plus un prix au kilomètre multiplié par la distance parcourue. Le forfait est souvent la recette principale d'une commande.

A l'aide de ces informations, la nomenclature du modèle est D-PDPTW, problème dynamique de ramassage et livraison avec fenêtre à temps. Le tableau suivant récapitule le résultat de l'analyse.

Tableau 8 Résumé des caractéristiques du problème VRP considéré

Facteur	Caractéristique
Problème	Problème de ramassage et livraison avec fenêtre a temps
Dynamisme	Élevé
Fenêtre a temps	Souple ou dure
Classe de ramassage	« One-to-one » chaque commande a un lieu de commande (restaurant) et livraison (client)
Nomenclature	D-PDPTW

3.3 Modélisation du problème :

Le problème représente un système de distribution de bien entre des lieux (souvent nommé nœuds) de commande vers des lieux de livraison. Les véhicules sont présents sur d'autre lieux (nommé souvent dépôt). La figure suivante présente un schéma de livraison réalisé par un livreur.

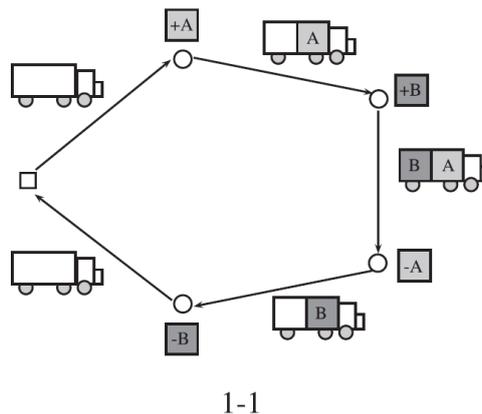


Figure 20 Les caractéristiques des problèmes one-to-one(Toth and Vigo, 2014)

3.3.1 Formulation :

On note n le nombre de commande à servir dans notre problème. Le problème peut être défini comme un graphe (N, A) , ou $N = P \cup D \cup \{0, 2n + 1\}$, $P = \{1, \dots, n\}$ et $D = \{n + 1, \dots, 2n\}$. P et D représentent les nœuds de ramassage et livraison, respectivement. 0 et $2n + 1$ représente le dépôt d'origine et de destination du livreur. A l'aide de cette représentation

chaque utilisateur i ou $1 \leq i \leq n$ associé à un nœud de ramassage i a une destination de ramassage $i + n$.

K Représente l'ensemble de véhicule disponible pour faire une livraison. Chaque véhicule $k \in K$ a une capacité Q_k et un temps global de route n'excédant pas T_k . Chaque nœuds $i \in N$ est associé a un volume de ramassage q_i et un temps de service d_i ou $q_0 = q_{2n+1} = 0$, $q_i = -q_{n+i}$ et $d_0 = d_{2n+1} = 0$. Une fenetre a temps $[e_i, l_i]$ est associé a chaque nœuds $i \in N$ ou e_i et l_i representent le plus tot et le plus tard temps de possibilité de réalisation de service, respectivement. $(i, j) \in A$ est une arête associée à une distance c_{ij} et un temps de voyage de t_{ij} . L dénote le maximum temps d'une course d'un livreur.

Pur chaque arête $(i, j) \in A$ et chaque véhicule $k \in K$, $x_{ij}^k = 1$ si un véhicule k traverse d'un nœud i vers un nœud j . Pour chaque nœud $i \in N$ et véhicule $k \in K$, B_i^k represente le temps où k commence son service au nœud i , et Q_i^k représente la charge portée par k après une visite du nœud i .

Le PDPTW peut être formulé à l'aide de ce programme linéaire :

$$\text{Min} \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{ij}^k x_{ij}^k$$

S.C :

$$\sum_{k \in K} \sum_{j \in N} x_{ij}^k = 1 \quad \forall i \in P \quad (1)$$

$$\sum_{j \in N} x_{ij}^k - \sum_{j \in N} x_{n+i,j}^k = 0 \quad \forall i \in P, k \in K \quad (2)$$

$$\sum_{j \in N} x_{0j}^k = 1 \quad \forall k \in K \quad (3)$$

$$\sum_{j \in N} x_{ji}^k - \sum_{j \in N} x_{ij}^k = 0 \quad \forall i \in P \cup D, k \in K \quad (4)$$

$$\sum_{j \in N} x_{i,2n+1}^k = 1 \quad \forall k \in K \quad (5)$$

$$B_j^k \geq (B_i^k + d_i + t_{ij})x_{ij}^k \quad \forall i \in N, j \in N, k \in K \quad (6)$$

$$Q_j^k \geq (Q_i^k + q_i)x_{ij}^k \quad \forall i \in N, j \in N, k \in K \quad (7)$$

$$L_i^k = B_{n+i}^k - (B_i^k + d_i) \quad \forall i \in P, k \in K \quad (8)$$

$$B_{2n+1}^k - B_0^k \leq T_k \quad \forall k \in K \quad (9)$$

$$e_i \leq B_i^k < l_i \quad \forall i \in N, k \in K \quad (10)$$

$$t_{i,n+i} \leq L_i^k \leq L \quad \forall i \in P, k \in K \quad (11)$$

$$\max\{0, q_i\} \leq Q_i^k \leq \min\{Q_k, Q_k + q_i\} \quad \forall i \in N, k \in K \quad (12)$$

$$x \in \{0,1\} \quad \forall i \in N, j \in N, k \in K \quad (13)$$

La fonction objective (1) minimise la distance de transport. Les contraintes (2) et (3) assure que chaque requête est servie exactement une seule fois et que chaque commande et destination doit être visitée par le même véhicule. Contraintes (4)-(6) garantissent que chaque route réalisée par un livreur doit commencer dans un dépôt d'origine et finir dans un autre dépôt

de destination. La continuité du temps et de charge est assurée par les contraintes (7) et (8). L'inégalité (9) définit le temps d'une course pour chaque livreur, cette contrainte est limitée par la (12) contrainte. Ces deux dernières s'assurent que chaque commande est réalisée avant une livraison, car cette contrainte assure la non négativité de L_i^k . L_i^k Assure que chaque nœud i est visité avant $n + i$ pour chaque utilisateur i . L'inégalité (10) limite la durée de chaque route. (11) et (13) imposent les fenêtres à temps et les contraintes de capacité, respectivement.

En pratique, résoudre ce problème nous permet d'affecter un ensemble de commandes reçus à un ensemble de livreur, en minimisant la distance parcourus par ces derniers.

3.3.2 Approches de résolution du problème :

En pratique, résoudre le problème à l'aide des méthodes exactes reste impossible. Si les affectations sont mises à jour après la réception d'une nouvelle commande, l'affectation des problèmes risques de prendre beaucoup de temps, à cause de la complexité des problèmes NP-difficile. Pour cela, les problèmes de livraisons dynamiques sont souvent résolus par des approches heuristiques.

Même si les problèmes de livraison de restaurant sont des problèmes récemment étudiés en milieu scientifique, il existe principalement deux approches pour résoudre les problèmes de livraison dynamiques. Le tableau suivant montre les projets menés ainsi qu'une brève description des approches.

Tableau 9 Articles de recherche abordant le problème de livraison

Articles de recherche	Description de l'approche
(Chen et al., 2020) réalisé avec l'équipe Data Science de Meituan.	La solution suggérée divise le problème en deux parties. La première partie consiste à trouver un livreur qui va prendre la commande. Ensuite, optimiser le chemin du livreur qui va assurer la commande.
(Steever et al., 2019)	Après la réception d'une commande, elle est affectée théoriquement à chaque livreur pour déterminer son chemin optimal. Le livreur qui va prendre la commande en réalité est celui qui minimise au plus la fonction objective.

L'approche utilisée pour résoudre cette problématique de livraison est l'approche utilisée par (Steever et al., 2019). Les exigences de Yassir Express ainsi que ses caractéristiques justifient principalement ce choix. Ces caractéristiques sont :

1. Attendre d'un nouveau livreur est impossible : retarder l'affectation d'une livraison pour attendre un client est impossible à Yassir Express. Le nombre de livreur est petit par rapport à d'autre entreprise (Meituan), d'où le risque d'attendre beaucoup de temps avant de livrer une commande.
2. Difficulté à gérer l'annulation de l'affectation de livraison : si un livreur décide de ne pas accepter une commande, l'approche non utilisée ne permettent pas de proposer

à un autre livreur cette commande en un temps constant. Il est nécessaire de refaire la résolution sans la présence de ce livreur.

Par contre, l'approche utilisée par (Steever et al., 2019) permet d'y remédier à ces problèmes. Cette approche est donc utilisée pour résoudre le problème.

3.3.3 Méthode de résolution :

Supposons qu'on a un ensemble de livreur localisé à un instant t sur un ensemble de dépôts nommé X . Ces dépôts sont souvent fictifs, car, en réalité, le livreur peut être près d'un restaurant ou bien il a juste mentionné sa présence à l'aide de l'application de Yassir Express.

A cet instant t . Chaque livreur possède déjà d'une route, cette route peut être vide (livreur à l'arrêt) ou contient un ensemble d'emplacement de commande et livraison. A cet instant, la fonction objective contient une distance globale théorique parcourue par l'ensemble des livreurs. L'algorithme peut déclencher s'il reçoit une nouvelle commande d'un client. La commande contient l'endroit de commande et de livraison émis à l'aide du système GPS du client.

L'approche de l'algorithme est simple. Elle suppose que chaque livreur va recevoir la commande. Supposons qu'un livreur x reçoit la commande, pour minimiser la distance parcourue par ce livreur, la route actuelle du livreur est modifiée tout en gardant le critère de ramassage des commandes précédentes. Cette procédure est réalisée pour l'intégralité des livreurs disponibles.

Après la détermination du chemin optimal pour chaque livreur. L'algorithme affecte à cette étape la commande au livreur qui minimise l'augmentation de la fonction objective.

En résumé, l'approche influencée par les travaux de (Steever et al., 2019) résout le problème de voyageur de commerce avec ramassage et livraison, ou TSPPD (*Traveling Salesman Problem with Pickup and Deliveries*) pour tous les livreurs. L'approche utilisée par l'algorithme est donc la suivante :

1. Réception d'une commande à l'instant t
2. L'Algorithme reçoit ainsi l'ensemble des lieux de ramassage et livraison réalisé par les livreurs à l'instant t .
3. L'algorithme calcule une matrice de distance entre les lieux de ramassage ,de livraison et des livreurs
4. Résoudre un problème TSPPD pour chaque livreur. Les données utilisées pour résoudre le problème sont la nouvelle commande et les commandes affecté précédemment au livreur.
5. La nouvelle commande est affectée au livreur qui augmente le moins la fonction objective, ce livreur est appelé le livreur optimal. Le chemin optimal est ainsi retourné par l'algorithme.
6. Enregistrer les meilleures solutions et les classer pour les utiliser si le livreur optimal refuse d'accepter cette commande (étape optionnelle)

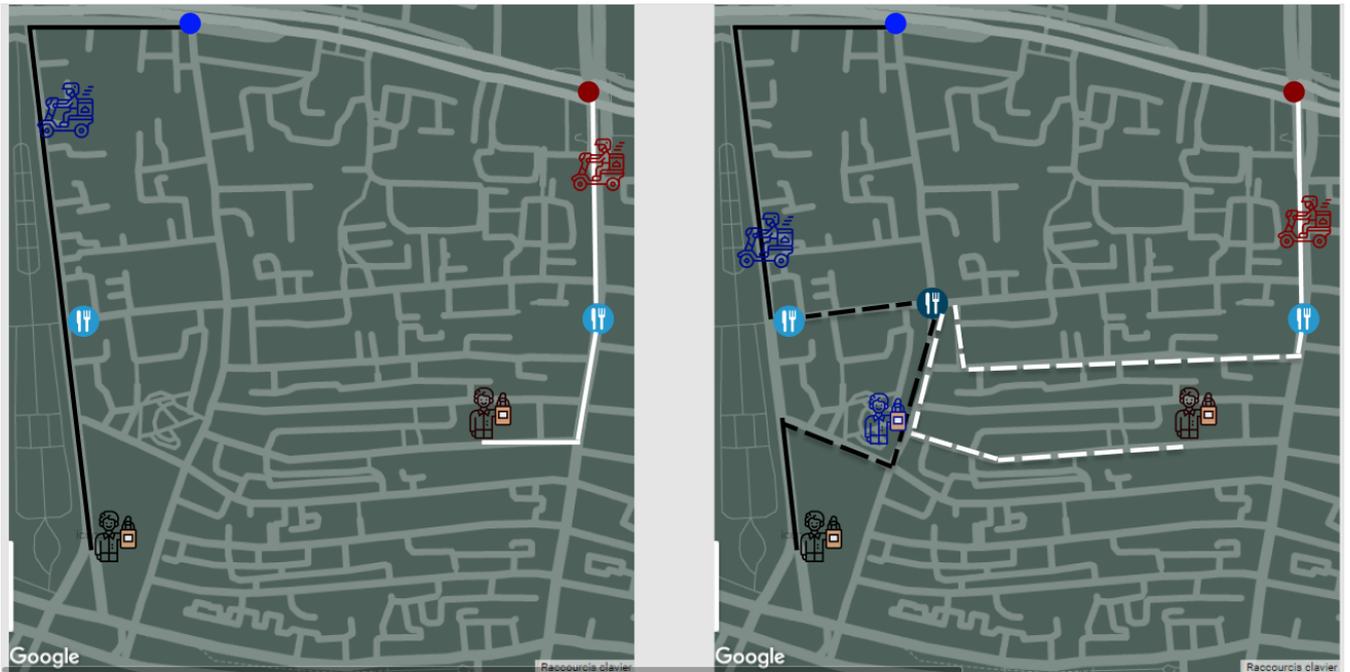


Figure 21 A gauche l'image représente la situation du système de livraison avant la réception d'une nouvelle commande. A droite, Lors de la réception, l'algorithme calcule le chemin optimal de chaque livreur

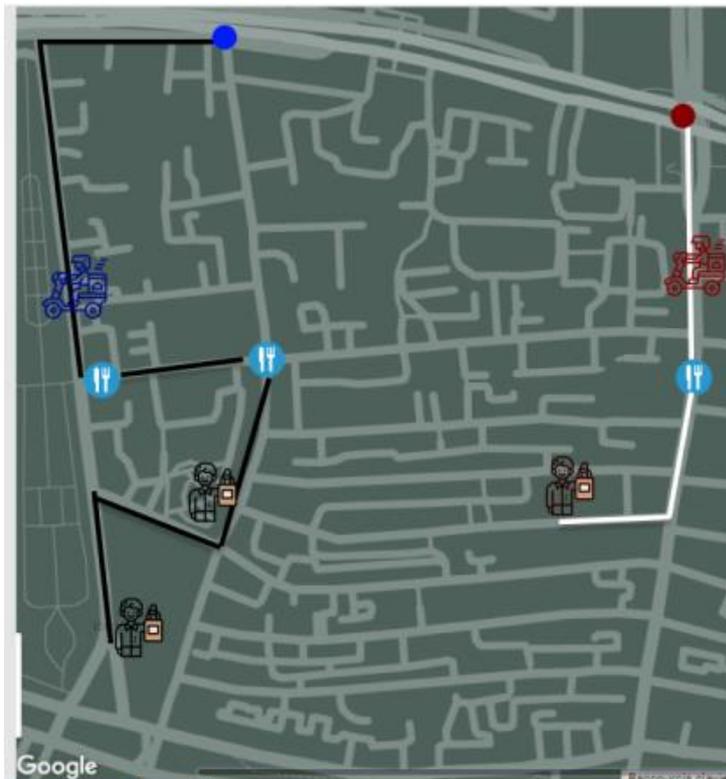


Figure 22 l'affectation de la nouvelle commande

3.3.4 Résolution du TSPPD :

Même si le sous-problème exige moins de computation par rapport à une résolution globale, la nécessité de résoudre ce sous-problème pour chaque livreur peut rendre le temps de computation long. De ce fait, une approche heuristique est utilisée pour résoudre ce problème. L'heuristique choisie est le système de colonies fourmis, plusieurs raisons justifient ce choix :

- 1) L'approche constructive facilite la satisfaction le critère de précédence : puisque le livreur n'a pas le droit de passer par un client sans avoir ramasser sa commande, assurer le critère de précédence pour les autres heuristiques reste une tâche difficile. Or, à l'aide de l'approche constructive de l'ACF, il est possible d'assurer ce critère.
- 2) La complexité d'autre méthode : implémenter les autres approches heuristiques en satisfaisant le critère de précédence rend ces dernières plus complexes par rapport à l'ACF. En termes de complexité algorithmique.

3.3.4.1 Système de colonies de fourmis modifié :

Le système de colonies de fourmis modifié est similaire à l'approche décrite dans le deuxième chapitre. Cependant, la différence existe sur le critère de choix des villes à visiter. L'ACF modifié est résumé sur la figure suivante et le tableau décrit le rôle de chaque constante.

Pour trouver le chemin optimal, l'algorithme commence par construire une route pour chaque fourmi dans une colonie. $N = U \cup P \cup D \cup \{0\}$ représente l'ensemble des villes que doit visiter une fourmi. 0 Représente le point de départ vers les autres destinations et n représente le nombre de commande pas encore ramassé et livré. $P = \{l + 1, \dots, l + n\}$ contient l'ensemble des lieux de ramassage. $D = \{(l + 1) + n, \dots, l + 2n\}$ représente l'ensemble des lieux de livraison. $U = \{1, 2, \dots, l\}$ est l'ensemble des lieux de livraison ramassé précédemment par le livreur. L'avantage principale de cette présentation est la possibilité de savoir si un nœud est de ramassage ou livraison et de connaître le lieu de livraison à partir d'un nœud de ramassage. Si $i \in P$, il est clair que son lieu de livraison est $i + n$; cette approche aide l'algorithme à trouver les lieux de livraison de chaque nœud de ramassage et vice versa.

Tableau 10 Description des différentes constantes de l'algorithme modifié

Constante	Description
n_iteration	Le nombre de fois ou la résolution du problème est faite. La résolution du problème plusieurs fois permet d'explorer plusieurs solution
n_fourmis	Le nombre de fourmis qui vont explorer les routes dans l'algorithme ACF. Ce chiffre représente la taille de la colonie
n_meilleur	Dans l'ACF modifié, seules les meilleures routes réalisées par les meilleures fourmis peuvent laisser une quantité de phéromone, car plusieurs parties de ces chemins font probablement parties du chemin optimal
défaillance	La défaillance représente le taux d'évaporation de phéromone après chaque passage des fourmis
Alpha et béta	Alpha et béta sont les constantes qui aident à calculer la densité de probabilité pour choisir la prochaine destination à visiter par l'algorithme . $p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha * \eta_{ij}^\beta}{\sum_{j \in N_i^k} \tau_{ij}^\alpha * \eta_{ij}^\beta} , j \in N_i^k \\ 0 \text{ sinon} \end{cases}$

Données : D : Matrice des distances ($n * n$)
 n : Nombre de nœuds n ($n = |U \cup P \cup D \cup \{0\}|$)
 N : ensemble de nœuds ($U \cup P \cup D \cup \{0\}$)
 n_fourmi , $n_meilleur$, $n_iteration$, défaillance, α , β : constantes
 l : nombre de client à livrer sans ramassage
Valeur retournées : le meilleur chemin construit par l'algorithme de colonie de fourmis modifié

```

i ← 0
j ← 0
m ← 0
meilleur_route ← {}

tant que i < n_iteration
    routefourmi ← {}

    tant que j < n_fourmis
        routefourmi ← {}
        impossibleAvisiter ← {D ∪ routefourmi}
        pheromone[n][n] ← initialiser_matrice_carré(D)
        precedent ← route_livreur[0]

        tant que m < n
            suivant ← choisir_prochaine_ville( N,D[precedent],
                impossibleAvisiter,pheromone[precedent])
            route ← route ∪ {suivant}

            si nœuds_ramassage(suivant) == vrai
                impossibleAvisiter ← deverrouiller_nœuds(suivant)

            impossibleAvisiter ← {impossibleAvisiter ∪ suivant}
            suivant ← precedent
            m ← m+1

        route_fourmi ← route_fourmi ∪ {route}
        j ← j+1
    route_fourmi ← classer_route_fourmi(route_fourmi)
    pheromone[n][n] ← placer_pheromone(route_fourmi, n_meilleur, défaillance)

    si qualité(meilleur_route) < qualité(route_fourmi[0])
        meilleur_route ← route_fourmi[0]

    i ← i+1

```

Figure 23 Méthodes de génération des routes

Il est clair qu'au début la fourmi ne peut pas choisir D et 0 , car aucun nœud de ramassage n'a été visité et il est impossible de visiter une ville deux fois. Une fois ces valeurs initialisées, l'algorithme choisi la prochaine ville à visiter. N contient l'ensemble des villes pour

résoudre le problème, **D[précédent]** et **phéromone[phéromone]** représentent les distances à partir de la ville précédemment visitée et la quantité de phéromone déposée, respectivement, et **impossibleAvisiter** contient une liste de nœuds que l'algorithme n'a pas le droit de sélectionner à cet instant. Ces nœuds sont impossibles à visiter pour deux raisons :

- Nœuds de ramassage n'est pas visité à cette phase de construction
- La ville est choisie durant les boucles précédentes.

Choisir_prochaine_ville est une fonction qui choisit un nœud à visiter par une fourmi. il est clair que, **D[précédent]** et **pheromone[phéromone]** sont des vecteurs. **Choisir_prochaine_ville** crée une densité de probabilité pour les nœuds possibles à visiter par une fourmi durant cette étape, selon la formule suivante :

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha * \eta_{ij}^\beta}{\sum_{j \in N_i^k} \tau_{ij}^\alpha * \eta_{ij}^\beta}, & j \notin impossible \ a \ visiter, \ i = \text{précédent} \\ 0 & \text{sinon} \end{cases}$$

Après le choix d'une ville à visiter par l'algorithme. Cette ville est ajoutée à la route de la fourmi. Si le nœud choisi est un nœud de ramassage, la livraison au client est possible. de ce fait, l'emplacement de livraison doit être enlever des nœuds **impossibleAvisiter**, c'est le rôle de la fonction **déverrouiller_nœuds**.

Puisqu'une fourmi prend un nœud pour chaque itération, cette partie de l'algorithme doit être répéter **n** fois, où **n** est le nombre total de ville à visiter. Cette approche permet de collecter l'intégralité des routes établies par les fourmis. La modification de l'ACF est principalement la suivante :

« Seuls les meilleures meilleur fourmis place du pheromone sur le chemin, le nombre de fourmis est la constante **n_meilleur** »

Cette approche permet une convergence rapide vers la solution optimale. La classification des solutions est réalisée par **classer_route_fourmis** ; cette fonction retourne la classification des routes selon la distance parcourue par les fourmis. La distance représente la fonction objective de l'élément. **Placer_pheromone** place le pheromone des meilleures routes et seuls **n_meilleur** fourmis réalisent cette tâche.

L'algorithme à la fin mesure la qualité (fonction objective) d'une solution trouvée par les fourmis, si cette solution est la meilleure solution à cette étape, le chemin optimal sera modifié.

3.3.4.2 Complexité de la solution :

A l'aide de la présentation des matrices des distances et la modification de l'algorithme. La complexité de l'algorithme est de l'ordre $O(n * n_{fourmi} * n_{iteration})$ sont les trois boucles. Cependant, à cause de la répétition de cette solution pour chaque livreur. Il est préférable de fixer des constantes qui rendent la convergence rapide, sans être coincé dans un optimum local.

En pratique plusieurs approches sont utilisés pour limiter le temps de résolution du problème. Ces deux approches vont être utilisées en pratique pour réduire la complexité du sous-problème :

- Limiter le nombre de commande à planifier pour un livreur : cette limitation réduit automatiquement **n**, le nombre de commande à planifier
- Minimiser le nombre d'itération : limiter le nombre d'itération a l'inconvénient de réduire la qualité de la solution optimale. Cependant, pour de petit instant de **n**, cette approche n'affecte pas la solution car l'ensemble des routes à explorer est réduit.

3.3.5 Implémentation de la solution du TSPPD :

Dans cette partie, une simulation est réalisée pour trouver la solution a un problème TSPPD. Cependant, une hypothèse est réalisée pour le livreur. On suppose que le livreur a une limite sur le nombre de commande qui peut porter sur son chemin. Cette hypothèse est instaurée car, en pratique, un livreur a une box de volume limité.

Pour réaliser la simulation, plusieurs simulateurs sont développés par des entreprises ou même en open source. Le tableau suivant résume quatre approches pour résoudre le problème. Ce tableau est loin d'être exhaustive sur les outils informatiques disponibles pour résoudre le sous-problème.

Tableau 11 description de plusieurs solveurs d'optimisation (moi-même)

Approche et outil	Description	Inconvénient
CPLEX	Solveur développé par IBM pour résoudre les problèmes d'optimisation statiques. L'approche utilisée est une approche exacte, similaire à Branch and Cut ou Branch and Bound. Le logiciel utilise un langage de programmation mathématique appelé <i>Object Programming language</i>	<ul style="list-style-type: none"> • Cout d'acquisition pour les entreprises • Solveur rarement utilisé pour les problèmes dynamiques • L'utilisation d'approche exacte rend le temps d'exécution d'une solution relativement long
Google Or-Tools	Solveur open source crée par Google pour résoudre les problèmes d'optimisation, particulièrement les problèmes VRP, PDP ...L'approche utilisée pour résoudre les problèmes est une combinaison de l'approche recherche locale et recuit simulé. Ce logiciel est développé en C++	<ul style="list-style-type: none"> • Le solveur ne dispose pas de l'approche ACF. • Ce solveur n'est pas adapté pour résoudre des problèmes dynamiques.
Jspirit	Solveur open source développé par Graphhopper ,GmbH .Jspirit est similaire a Or-Tools en approche de résolution et fonctionnalité. Cependant, le solveur est développé en Java.	<ul style="list-style-type: none"> • Le solveur ne dispose pas de l'approche ACF. • Ce solveur n'est pas adapté pour résoudre des problèmes dynamiques.
Python	Ce langage de programmation n'est pas un solveur. Cependant, une solution peut être développée pour tous les problèmes liés en VRP. Il est possible de programmer l'ACF, mais avec une difficulté.	<ul style="list-style-type: none"> • Temps de développement d'une solution est en général plus longs par rapport aux solveurs • Difficulté à modéliser les problèmes VRP

Sur ce tableau, on constate que les solveurs ne sont généralement pas conçus pour résoudre des problèmes dynamiques. De plus, les solveurs n'ont pas l'approche ACF pour résoudre les problèmes. Python est donc choisi pour résoudre le problème TSPPD.

3.4 Préparation des données :

Les données du tableau suivant représentent la matrice des distances des lieux que doit visiter un livreur. La notation suivie est celle décrite dans la section précédente. Plusieurs hypothèses pour créer la matrice de distance :

- Les fenêtres à temps sont suffisamment larges pour ne pas les prendre en considération dans le modelé mathématique
- Un livreur ne peut prendre que 3 commandes à la fois. Prendre plus de trois commandes risque d'encombrer la « box » du livreur
- On suppose qu'à l'instant du début de l'algorithme. Le livreur porte déjà deux commandes à livrer vers l_1 et l_2
- Si i et j sont deux nœuds et r_{ij} la distance de i vers j , $r_{ij} = r_{ji}$. La matrice des distances est donc symétrique.
- Chaque commande est associée à un seul point de ramassage p_i et un unique point de livraison d_i
- L'algorithme ne prend pas en considération la distance retour vers le nœud de départ du livreur dans la fonction objective.

Tableau 12 Matrices des distances du problème a resoudre

Indices (Python)	0	1	2	3	4	5	6	7	8	9	10	11	12
Nœuds	0	l_1	l_2	p_1	p_2	p_3	p_4	p_5	d_1	d_2	d_3	d_4	d_5
0	0	2451	713	1018	1631	1374	2408	213	2571	875	1420	2145	1972
l_1		0	1745	1524	831	1240	959	2596	403	1589	1374	357	579
l_2			0	355	920	803	1737	851	1858	262	940	1453	1260
p_1				0	700	862	1395	1123	1584	466	1056	1280	987
p_2					0	663	1021	1769	949	796	879	586	371
p_3						0	1681	1551	1765	547	225	887	999
p_4							0	2493	678	1724	1891	1114	701
p_5								0	2699	1038	1605	2300	2099
d_1									0	1744	1645	653	600
d_2										0	679	1272	1162
d_3											0	1017	1200
d_4												0	504
d_5													0

3.4.1 Implémentation à l'aide de python :

Le programme python est divisé en deux classes. La première classe, appelé *MatrixColony*, gère la matrice des distances et rend le point de livraison associé à un point de départ, l'implémentation complète est disponible en annexe.

La première classe, *MatrixColony* aide à gérer la matrice des distances. Les fonctions principales sont celles qui retournent le lieu de livraison d_i équivalent à un lieu de ramassage p_i et vice versa.

La deuxième classe, *AntColony*, utilise l'approche décrite par l'algorithme théorique. Cette classe utilise les méthodes et attributs de la classe *MatrixColony* pour résoudre le problème TSPPD. Les figures suivantes résument les méthodes les plus importantes sur cette classe.

Les valeurs des constantes de l'ACF modifié sont prédéfinies après plusieurs expérimentations afin de minimiser le temps d'exécution du programme et maximiser la qualité de la solution. Cependant, cette configuration peut donner de mauvais résultats pour d'autres configurations d'une matrice de distance. Le tableau suivant montre la configuration initiale utilisée dans cette simulation.

Variable sur python	Variable sur l'algorithme théorique	Valeur par défaut
n_ants	n_fourmis	6
n_best	n_meilleur	3
n_iterations	n_iteration	10
decay	défaillance	0.9
alpha	alpha	1
beta	beta	4

```
def gen_path(self):
    pickup = self.distances.pickup_nodes
    route = np.array([0])
    prev = route[0]
    next_node = 0
    demands = self.distances.carried_demands
    pickup_selected = np.array([], dtype=int)
    locked_deliveries = self.distances.delivery_nodes
    cost = 0
    for i in range(self.distances.nodes - 1):
        locked_deliveries = self.get_locked_deliveries(pickup_selected)
        if (demands < 3):
            next_node = self.choose_node(
                self.pheromone[prev], self.distances.distance_matrix[prev],
                np.concatenate((route, locked_deliveries)))
        else:
            next_node = self.choose_node(
                self.pheromone[prev], self.distances.distance_matrix[prev],
                np.concatenate((pickup, route, locked_deliveries)))
        if (self.distances.is_pickup_node(next_node)):
            pickup_selected = np.append(pickup_selected, next_node)
            demands += 1
        else:
            demands -= 1
        route = np.append(route, next_node)
        cost += self.distances.distance_matrix[prev][next_node]
        prev = next_node
    return route, cost
```

Figure 24 Méthode de génération d'une route sur Python

```

def run(self):
    shortest_path = None
    all_time_shortest_path = ("placeholder", np.inf)
    for i in range(self.n_iterations):
        all_paths = self.gen_all_paths()
        self.spread_pheronome(all_paths,
                              self.n_best,
                              shortest_path=shortest_path)
        shortest_path = min(all_paths, key=lambda x: x[1])
        if shortest_path[1] < all_time_shortest_path[1]:
            all_time_shortest_path = shortest_path
        self.pheromone = self.pheromone * self.decay
    return all_time_shortest_path

```

Figure 25 La méthode qui résout le sous-problème

Pour dérouler le programme il suffit juste d'introduire la matrice de distance ainsi que le nombre de commande portée par le livreur à cet instant.

```

if __name__ == "__main__":
    data = {}
    data['distance_matrix'] = np.array([
        [0, 2451, 713, 1018, 1631, 1374, 2408, 213, 2571, 875, 1420, 2145, 1972],
        [2451, 0, 1745, 1524, 831, 1240, 959, 2596, 403, 1589, 1374, 357, 579],
        [713, 1745, 0, 355, 920, 803, 1737, 851, 1858, 262, 940, 1453, 1260],
        [1018, 1524, 355, 0, 700, 862, 1395, 1123, 1584, 466, 1056, 1280, 987],
        [1631, 831, 920, 700, 0, 663, 1021, 1769, 949, 796, 879, 586, 371],
        [1374, 1240, 803, 862, 663, 0, 1681, 1551, 1765, 547, 225, 887, 999],
        [2408, 959, 1737, 1395, 1021, 1681, 0, 2493, 678, 1724, 1891, 1114, 701],
        [213, 2596, 851, 1123, 1769, 1551, 2493, 0, 2699, 1038, 1605, 2300, 2099],
        [2571, 403, 1858, 1584, 949, 1765, 678, 2699, 0, 1744, 1645, 653, 600],
        [875, 1589, 262, 466, 796, 547, 1724, 1038, 1744, 0, 679, 1272, 1162],
        [1420, 1374, 940, 1056, 879, 225, 1891, 1605, 1645, 679, 0, 1017, 1200],
        [2145, 357, 1453, 1280, 586, 887, 1114, 2300, 653, 1272, 1017, 0, 504],
        [1972, 579, 1260, 987, 371, 999, 701, 2099, 600, 1162, 1200, 504, 0],
    ], dtype = float) # yapf: disable
    data['demands'] = 2
    np.fill_diagonal(data['distance_matrix'], np.inf)
    matrix = MatrixColony(data['distance_matrix'], data['demands'])

    print(matrix.delivery_nodes)
    print(matrix.pickup_nodes)

    antcolony = AntColony(matrix)
    print(antcolony.run())

```

Figure 26 Insertion des caractéristiques sur Python

Une fois la configuration utilisée, les résultats suivants sont obtenus :

```

noeuds de livraison :
[ 8  9 10 11 12]

noeuds de ramassage :
[3 4 5 6 7]

noeuds a livrer seulement :
[1, 2]

le chemin optimal du livreur
[ 0  7  2  3 12  6  8  1 11  4  9  5 10]

la distance totale parcourue:
6699.0

```

Figure 27 Résultat obtenu

Après plusieurs itérations de l'algorithme, la solution optimale trouvée est 6998. Cette solution est probablement optimale, mais puisque l'approche est heuristique et la complexité du problème est élevée, il peut y exister de meilleur résultat.

3.4.2 Justification des constantes prédéfinies:

Pour justifier les constantes définies sur l'algorithme doivent être analysés pour déterminer les valeurs optimales pour résoudre le sous-problème. Le but de trouver les valeurs optimales est de minimiser le temps d'exécution de l'algorithme tout en maximisant la possibilité de trouver l'optimum du problème.

Cependant plusieurs remarques doivent être établies :

- **α est égale à 1** : l'augmentation de α rend les décisions prises dépendantes sur les solutions précédentes. Or, les solutions précédentes sont en générale moins optimales que les solutions qu'on souhaite trouver. α doit garder donc une valeur minimale
- **n_ants, n_iteration, n_best sont des variables à minimiser** : ces constantes augmentent la complexité du problème. Les minimiser rend l'algorithme plus rapide en exécution, mais aussi moins fiable. Trouver un équilibre est donc important.
- **decay dépend de n_iteration** : Si pour chaque itération, un chemin (i, j) n'est pas choisi, la quantité de phéromone doit disparaître durant les dernières phases d'itération, car la probabilité que la route optimale ne contient pas le chemin (i, j) est supposé élevé. Pour assurer cette approche, **decay** est égale à 0.9

Pour déterminer la valeur des autres constantes, deux études ont été réalisées. Le but est de trouver des constantes fiables et optimales pour la résolution du problème.

3.4.2.1 La première étude :

Cette étude est réalisée sur la relation entre plusieurs valeurs de **beta** et **n_iteration** ; **beta** est un nombre entier $1 \leq beta \leq 5$. Cette étude résout 20 fois le problème pour chaque valeur de beta et affiche l'évolution moyenne des solutions pour chaque **n_iteration** ≤ 30 .

La figure suivante affiche le résultat de cette simulation sur un graphe. Deux constats peuvent être tirés de cette simulation :

- 4 et 5 sont les meilleures valeurs de **beta** : La convergence de la solution est rapide lorsque **beta** est égale à ces deux valeurs. La différence entre ces deux valeurs est faible.
- La convergence est moins rapide lorsque **n_iteration** ≥ 10 : Si 6998 est considéré comme la solution optimale. Après 10 itérations, l'algorithme donne une solution moyenne proche à 10% de la solution optimale, Pour beta entre 4 et 5.

A l'aide de cette étude, les constantes **beta** et **n_iteration** sont établies 4 et 10, respectivement. Ces valeurs assurent un temps de computation faible et une convergence rapide vers la solution optimale.

3.4.2.2 La seconde étude :

Le but de la seconde étude consiste à trouver la valeur optimale pour **n_meilleur**. Pour cela, **n_meilleur** est varié sur quatre expérimentations. Le tableau suivant résume les paramètres utilisés.

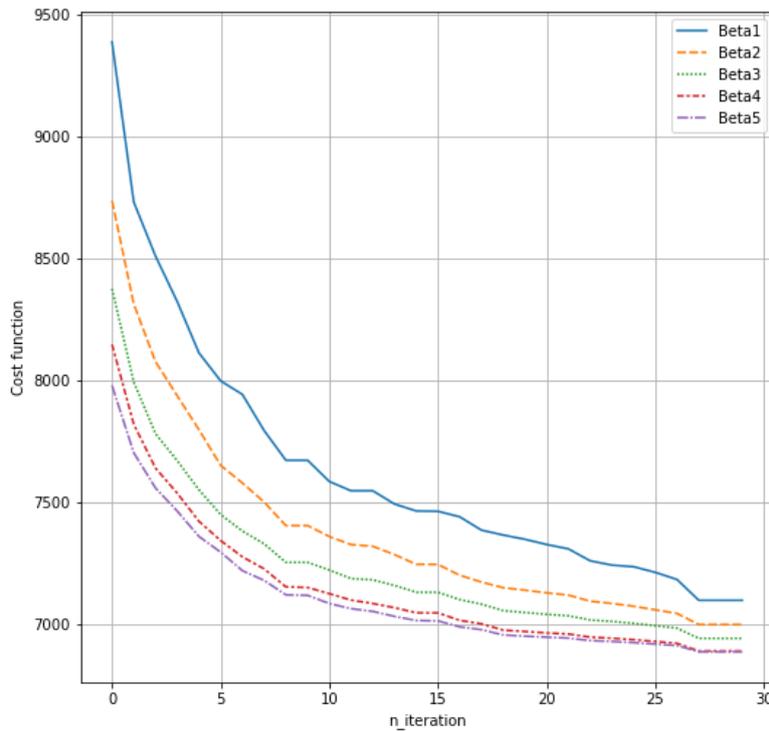


Figure 28 convergence de la solution optimale pour plusieurs itérations de l'algorithme

Tableau 13 Les constantes simulés durant la seconde etude

Variable python	Simulation 1	Simulation 2
n_ants	6	10
n_best	[1 ; n_ants]	[1 ; n_ants]
n_iterations	5	5
decay	0.9	0.9
alpha	1	1
beta	1	1

Pour chaque valeur de **n_best**, 50 déroulements est réalisé pour déterminer le meilleur résultat pour chaque itération. Les diagrammes à moustaches suivants montrent les variations de la distance parcourue par le chemin optimal.

Pour la première simulation, on remarque que les deux valeurs médianes les plus faibles sont celles établies lorsque **n_best** égale à 3 ou à 5. Cependant, la variation des valeurs lorsque **n_best** = 3 est plus faible que les valeurs pour 5. Il est clair que les meilleurs résultats sont en moyenne trouvés lorsque **n_best** = 3.

Lors de la seconde simulation, on remarque que les meilleurs résultats sont obtenus lorsque **n_best** égale à 5 ou à 8. Cependant, la variation des résultats est plus faible lorsque **n_best** égale à 5.

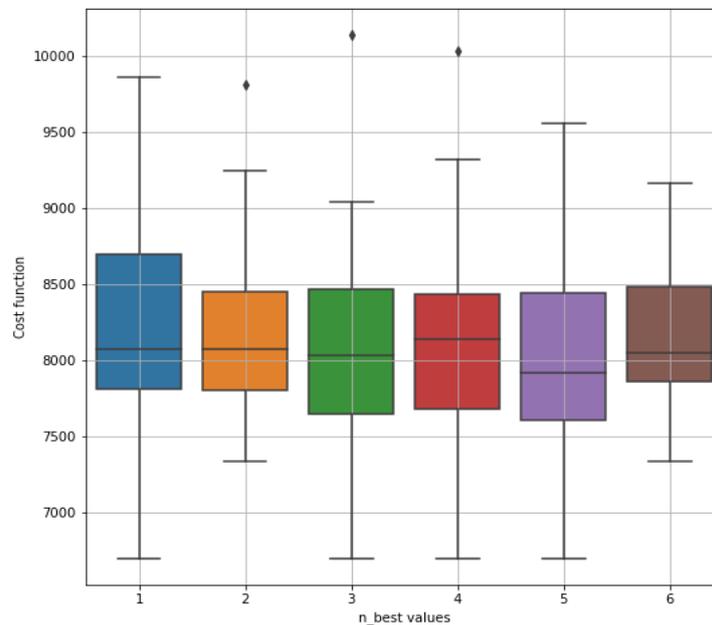


Figure 29 Résultat des coûts de transport sur la variation de **n_best** pour la première simulation

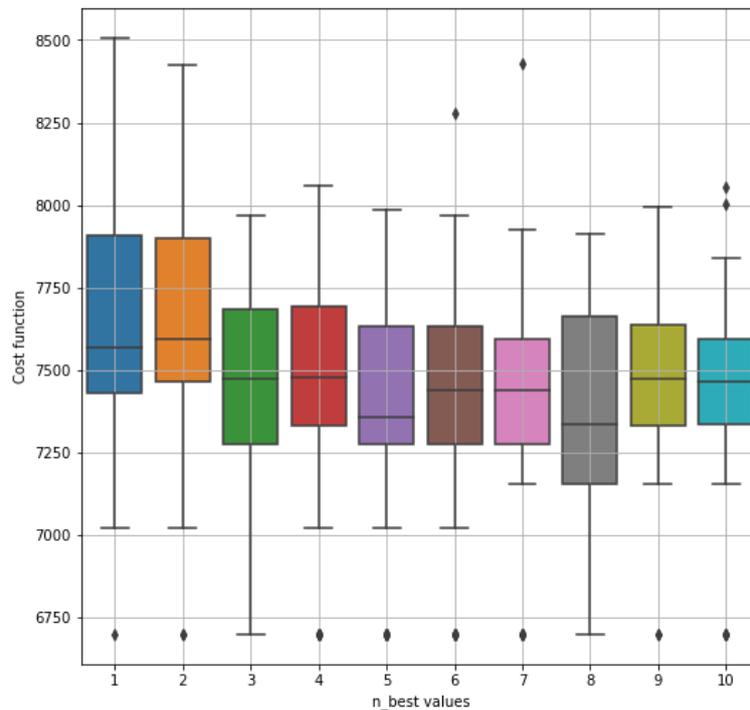


Figure 30 Résultat des coûts de transport sur la variation de **n_best** pour la seconde simulation

A l'aide de ces deux simulations, on remarque que les meilleurs résultats ont une probabilité plus élevée d'être obtenues lorsque la moitié des meilleurs chemins (fourmis) déposent de la phéromone. Mettre des phéromones sur la moitié

3.4.2.3 Conclusion des études :

Les deux dernières études ont permis de justifier en partie les constantes établies pour résoudre le sous-problème. Ces constantes peuvent être utilisées pour résoudre le sous-problème avec une faible complexité, donc résoudre le problème global rapidement.

3.5 Limites du modèle :

L'approche utilisée pour résoudre le problème d'optimisation présente plusieurs limitations pour adopter cette approche. Ces limitations sont à la fois technologiques, scientifique et commerciale.

3.5.1 Limites technologiques :

La limite technologique est principalement liée aux technologies liées à réaliser le projet. Les deux principales limites sont :

- La vitesse d'exécution de Python :
 - En réalité, Python est choisi pour permettre une implémentation rapide et relativement facile d'une solution. Cependant, il existe plusieurs langages de programmation qui ont un temps d'exécution plus rapide que Python, par exemple C, C++ et même récemment Rust. Un temps d'exécution plus rapide implique une génération de solution plus rapide. D'où une capacité de résoudre des problèmes plus complexes en moins de temps d'exécution.
- L'élaboration d'une matrice des distances :
 - En pratique, l'élaboration de matrice des distances nécessite l'utilisation d'un logiciel de routage tel GoogleMaps ou OpenRoutesService. En réalité, ces logiciels limitent la taille des matrices de distances à élaborer car cette tâche est coûteuse (en computation) pour les prestataires de services. Les prestataires de services limitent souvent le nombre de matrice de distance à invoquer chaque heure et la dimension de cette matrice. L'un des prochains objectifs consiste à trouver une approche pour pouvoir utiliser ces services en prenant en considération leurs critères.
- Performance limitée de la base de données :
 - L'approche de résolution nécessite un flux d'information important. Or, la base de données de Yassir ne peut pas supporter un flux de requête important pour tirer les données des livreurs et clients, chaque minute. L'amélioration de la base de données reste une tâche importante pour pouvoir implémenter la solution.

3.5.2 Limites scientifiques :

Les problèmes de ramassage et livraison dynamique sont des problèmes qui ont surgit très récemment. L'étude scientifique est particulièrement en retard dans la gestion d'annulation de commande par un livreur et la gestion de commande provenant de plusieurs emplacements de ramassage. De ce fait, les solutions proposés sont souvent pas optimales dans le cas réel et c'est le cas pour la solution adoptée.

De plus, l'approche développée peut fonctionner pour seulement un nombre faible de livreur, car chaque livreur augmente la complexité de la solution.

3.5.3 Limites commerciales :

Même si un système de paiement en ligne a émergé récemment en Algérie, rare sont les algériens qui l'ont adopté. Cette non-utilisation oblige à Yassir à ne pas automatiser plusieurs processus, car les clients doivent être obligatoirement appelé avant de faire la livraison.

3.6 Perspectives futures :

Ce travail a permis de modéliser et programmer une partie de l'approche proposé dans ce rapport. Une première piste d'amélioration consiste à programmer la solution en C++ ou Rust. Puisque l'algorithme est déterminé, programmer cette partie reste une tâche relativement plus simple.

Un deuxième champ d'amélioration consiste à créer une simulation pour étudier mieux les flux de transport. Ce simulateur va aider à mieux tester les solutions proposées avant de mettre le service en exploitation. Les simulateurs aident à mieux tester des cas réels et aussi de proposer des solutions innovantes sans avoir besoin de déployer une solution qui risque de perturber le fonctionnement du produit.

Le modèle développé pour résoudre le problème est correct. L'avantage principale de la modélisation utilisé est la possibilité d'optimiser plusieurs paramètres de l'entreprise. Une amélioration à faire dépend du besoin d'entreprise, par exemple économiser les frais de livraison en minimisant le nombre de livreur.

En parallèle, un besoin clair en recherche scientifique est l'absence de données publics pour étudier les problèmes dynamiques. Souvent, les chercheurs dans le domaine d'optimisation dynamiques utilisent des données d'entreprises privées. Cette approche rend la recherche scientifique dépendante sur plusieurs entreprises qui ne veulent pas publier les données. Cette dépendance rend aussi la comparaison des différentes approches publiées difficile.

De plus, les solutions proposées ne prennent souvent pas en considération des critères qui arrivent souvent, par exemple le refus par les livreurs de faire de commande n'est souvent pas pris en considération dans l'optimisation de la solution. Trouver ainsi des approches pour résoudre ce problème reste important.

Enfin, un futur projet important consiste à implémenter d'autres solutions trouvées sur des articles de recherches ou combiner plusieurs approches de solution. Cette approche aidera l'entreprise à optimiser encore plus la solution proposée.

3.7 Conclusion :

Dans ce dernier chapitre, une méthodologie d'automatisation du service de livraison de Yassir Express a été élaboré, de la sélection du véhicule du transport, jusqu'à l'optimisation de son chemin de livraison. Une implémentation d'une partie de l'approche a été également proposée pour résoudre le sous-problème.

A l'heure actuel, la programmation d'une solution globale est en réalisation pour valider la solution et l'utiliser non-seulement dans le service Express, mais aussi dans les autres services de Yassir, particulièrement le service Market.

La solution proposée se résume en deux étapes :

- La première étape consiste à collecter l'emplacement en temps réel des livreurs disponibles et leur commande pour faire une livraison.
- Trouver le livreur qui optimise au mieux la livraison d'une nouvelle commande.

Conclusion générale :

Le service de livraison à domicile a connu une forte évolution à cause d'un besoin client changeant à la pandémie récente. La concurrence sévère présente dans ce service a contraint les entreprises de livraison à optimiser la gestion des coûts liées au transport et opération. Yassir souhaite garder son statut de leader sur le marché algérien et maghrébin à travers l'utilisation des technologies de l'information et communication.

C'est dans le cadre de l'innovation que le projet d'automatisation de l'affectation du service livraison a été mis en place. L'entreprise souhaite minimiser les coûts liés au service de transport tout en satisfaisant les clients et en maximisant les gains des partenaires de Yassir.

Pour répondre cette problématique, un diagnostic global des services proposés par Yassir a été analysé. Ensuite, un diagnostic stratégique et interne pour tirer la problématique de l'entreprise. Cette analyse a aussi permis de suggérer une solution en accordance avec le besoin de l'entreprise.

Suite à cela, la deuxième partie du projet a été consacré à la définition de base de plusieurs approches scientifiques utilisées pour répondre à la problématique. Une revue bibliographique a été abordée sur :

- Le problème de tournées de véhicules (VRP)
- Les problèmes variantes du problème de tournées de véhicules
- La modélisation du problème de VRP avec ramassage et livraison
- L'explication des différentes approches de résolution exacts et métaheuristiques
- L'explication détaillé de l'approche approximative de colonies de fourmis

La revue terminée, le dernier chapitre regroupe les approches décrites précédemment pour résoudre la problématique énoncée en entreprise. En première partie, une modélisation mathématique a été abordée pour la résolution du problème en programmation mathématique. Ensuite, plusieurs approches de résolution ont été abordée et une partie de la solution a été programmé en Python.

Cette solution a reçu un écho favorable de la part du responsable Data de Yassir, car elle répond parfaitement à la demande émise par le service Data. Actuellement, les autres jalons sont en cours de réalisation avec les améliorations proposées sur ce projet, à savoir :

- La programmation de la solution en C++
- L'amélioration des bases de données pour supporter le flux d'information nécessaire pour réaliser la solution

Les domaines d'utilisation du VRP restent larges. Plusieurs problématiques en entreprise et même gouvernemental peuvent être résolu à l'aide de l'optimisation combinatoire. La gestion des déchets, la livraison de produits, le transport en commun représentent des domaines d'application clair d'optimisation. Dans un monde plus complexe, la bonne prise de décision passe par des approches d'optimisation.

Bibliographie :

- Abd-Alsabour, N., 2016. Hybrid Metaheuristics for Classification Problems, in: Ramakrishnan, S. (Ed.), Pattern Recognition - Analysis and Applications. InTech. <https://doi.org/10.5772/65253>
- Brulhart, F., Safari, an O.M.C., 2011. Les 7 points clés du diagnostic stratégique.
- Casquilho, M., n.d. Integer programming: the branch and bound method.
- Chan, D., Voortman, F., Rogers, S., 2018. The rise of the platform economy.
- Chen, J., Wang, S., Wang, L., Zheng, J., Cha, Y., Hao, J., He, R., Sun, Z., 2020. A Hybrid Differential Evolution Algorithm for the Online Meal Delivery Problem, in: 2020 IEEE Congress on Evolutionary Computation (CEC). Presented at the 2020 IEEE Congress on Evolutionary Computation (CEC), IEEE, Glasgow, United Kingdom, pp. 1–8. <https://doi.org/10.1109/CEC48606.2020.9185792>
- Chou, W., Li, I., Zhang, L., 2017. New Governance of the Platform Economy. Deloitte.
- Cordeau, J.-F., 2006. A Branch-and-Cut Algorithm for the Dial-a-Ride Problem. *Oper. Res.* 54, 573–586. <https://doi.org/10.1287/opre.1060.0283>
- Dorigo, M., Birattari, M., Stutzle, T., 2006. Ant colony optimization. *IEEE Comput. Intell. Mag.* 1, 28–39. <https://doi.org/10.1109/MCI.2006.329691>
- E-conomy Africa 2020, 2020. . Google and IFC, a member of the World Bank Group.
- Luke, S., 2013. Essentials of metaheuristics: a set of undergraduate lecture notes, Second edition, online version 2.0. ed. lulu.com, Morrisville, N.C.
- Marchand, H., Wolsey, L.A., 2001. Aggregation and Mixed Integer Rounding to Solve MIPs. *Oper. Res.* 49, 363–371. <https://doi.org/10.1287/opre.49.3.363.11211>
- Mitrović-Minić, S., Laporte, G., 2006. The Pickup And Delivery Problem With Time Windows And Transshipment. *INFOR Inf. Syst. Oper. Res.* 44, 217–227. <https://doi.org/10.1080/03155986.2006.11732749>
- Moazed, A., Johnson, N., 2016. Modern monopolies: what it takes to dominate the 21st-century economy, First edition. ed. St. Martin's Press, New York, N.Y.

Ralphs, T.K., Kopman, L., Pulleyblank, W.R., Trotter, L.E., 2003. On the capacitated vehicle routing problem. *Math. Program.* 94, 343–359. <https://doi.org/10.1007/s10107-002-0323-0>

Saidoun, N., 2020. *La vie d'une start-up racontée par Yassir*. Lib. Algér.

Steever, Z., Karwan, M., Murray, C., 2019. Dynamic courier routing for a food delivery service. *Comput. Oper. Res.* 107, 173–188. <https://doi.org/10.1016/j.cor.2019.03.008>

Toth, P., Vigo, D. (Eds.), 2014. *Vehicle routing: problems, methods, and applications*, Second edition. ed, MOS-SIAM series on optimization. Society for Industrial and Applied Mathematics : Mathematical Optimization Society, Philadelphia.

Annexes :

Annexe 1 : résolution du sous-problème a l'aide de python :

```
import numpy as np

class MatrixColony():
    def __init__(self, distance_matrix, carried_demands):
        self.distance_matrix = np.array(distance_matrix)
        self.carried_demands = carried_demands
        self.pickup_nodes = np.arange(self.carried_demands + 1, self.separation_point())
        self.delivery_nodes = np.arange(self.separation_point(), len(self.distance_matrix))
        self.initial_delivery_destinations = self.initial_delivery_locations()
        self.nodes = self.get_nodes()
        self.requests = self.delivery_nodes.shape[0]

    def initial_delivery_locations(self):
        if self.carried_demands >= 0:
            return np.arange(1, self.carried_demands + 1, 1)
        else:
            return []

    def get_nodes(self):
        return self.distance_matrix.shape[0]

    def separation_point(self):
        return int((self.get_nodes() + self.carried_demands + 1) / 2)

    def get_pickup_node(self, x):
        if (x.size == 0):
            return []
        if (self.separation_point() <= x < len(self.distance_matrix)):
            return self.pickup_nodes[x % self.separation_point()]

    def get_delivery_node(self, x):
        if (x.size == 0):
            return []
        if ((self.carried_demands < x).all() &
            (x < self.separation_point()).all()):
            return (x - self.carried_demands - 1)

    def is_pickup_node(self, x):
        if (self.carried_demands < x < self.separation_point()):
            return True
        else:
            return False
```

```

class AntColony():
    def __init__(self,
                 distances,
                 n_ants=5,
                 n_best=2,
                 n_iterations=10,
                 decay=0.9,
                 alpha=1,
                 beta=3):
        self.distances = distances
        self.pheromone = np.ones(self.distances.distance_matrix.shape
                                ) / self.distances.distance_matrix.shape[0]

        self.n_ants = n_ants
        self.n_best = n_best
        self.n_iterations = n_iterations
        self.decay = decay
        self.alpha = alpha
        self.beta = beta

    def gen_path(self):
        pickup = self.distances.pickup_nodes
        route = np.array([0])
        prev = route[0]
        next_node = 0
        demands = self.distances.carried_demands
        pickup_selected = np.array([], dtype=int)
        locked_deliveries = self.distances.delivery_nodes
        cost = 0
        for i in range(self.distances.nodes - 1):
            locked_deliveries = self.get_locked_deliveries(pickup_selected)
            if (demands < 3):
                next_node = self.choose_node(
                    self.pheromone[prev], self.distances.distance_matrix[prev],
                    np.concatenate((route, locked_deliveries)))
            else:
                next_node = self.choose_node(
                    self.pheromone[prev], self.distances.distance_matrix[prev],
                    np.concatenate((pickup, route, locked_deliveries)))
            if (self.distances.is_pickup_node(next_node)):
                pickup_selected = np.append(pickup_selected, next_node)
                demands += 1
            else:
                demands -= 1
                route = np.append(route, next_node)
                cost += self.distances.distance_matrix[prev][next_node]
                prev = next_node
        return route, cost

```

```

def get_locked_deliveries(self, pickup):
    indexes = self.distances.get_delivery_node(pickup)
    locked = np.delete(self.distances.delivery_nodes, indexes)
    return locked

def choose_node(self, pheromone, dist, locked_nodes):
    pheromone = np.copy(pheromone)
    pheromone[list(locked_nodes)] = 0
    row = pheromone**self.alpha * ((1.0 / dist)**self.beta)
    norm_row = row / row.sum()
    move = np.random.choice(range(len(self.distances.distance_matrix)),
                             1,
                             p=norm_row)[0]

    return move

def gen_all_paths(self):
    all_paths = []
    for i in range(self.n_ants):
        all_paths.append(self.gen_path())
    return all_paths

def spread_pheromone(self, all_paths, n_best, shortest_path):
    sorted_paths = sorted(all_paths, key=lambda x: x[1])
    for path, dist in sorted_paths[:n_best]:
        for i in range(1, self.distances.nodes):
            self.pheromone[path[i]][path[
                i - 1]] += 1.0 / self.distances.distance_matrix[path[i]][
                path[i - 1]]
            self.pheromone[path[i - 1]][
                path[i]] += 1.0 / self.distances.distance_matrix[path[
                i - 1]][path[i]]

def run(self):
    shortest_path = None
    all_time_shortest_path = ("placeholder", np.inf)
    for i in range(self.n_iterations):
        all_paths = self.gen_all_paths()
        self.spread_pheromone(all_paths,
                              self.n_best,
                              shortest_path=shortest_path)
        shortest_path = min(all_paths, key=lambda x: x[1])
        if shortest_path[1] < all_time_shortest_path[1]:
            all_time_shortest_path = shortest_path
        self.pheromone = self.pheromone * self.decay
    return all_time_shortest_path

```

```

if __name__ == "__main__":
    data = {}
    data['distance_matrix'] = np.array([
        [0, 2451, 713, 1018, 1631, 1374, 2408, 213, 2571, 875, 1420, 2145, 1972],
        [2451, 0, 1745, 1524, 831, 1240, 959, 2596, 403, 1589, 1374, 357, 579],
        [713, 1745, 0, 355, 920, 803, 1737, 851, 1858, 262, 940, 1453, 1260],
        [1018, 1524, 355, 0, 700, 862, 1395, 1123, 1584, 466, 1056, 1280, 987],
        [1631, 831, 920, 700, 0, 663, 1021, 1769, 949, 796, 879, 586, 371],
        [1374, 1240, 803, 862, 663, 0, 1681, 1551, 1765, 547, 225, 887, 999],
        [2408, 959, 1737, 1395, 1021, 1681, 0, 2493, 678, 1724, 1891, 1114, 701],
        [213, 2596, 851, 1123, 1769, 1551, 2493, 0, 2699, 1038, 1605, 2300, 2099],
        [2571, 403, 1858, 1584, 949, 1765, 678, 2699, 0, 1744, 1645, 653, 600],
        [875, 1589, 262, 466, 796, 547, 1724, 1038, 1744, 0, 679, 1272, 1162],
        [1420, 1374, 940, 1056, 879, 225, 1891, 1605, 1645, 679, 0, 1017, 1200],
        [2145, 357, 1453, 1280, 586, 887, 1114, 2300, 653, 1272, 1017, 0, 504],
        [1972, 579, 1260, 987, 371, 999, 701, 2099, 600, 1162, 1200, 504, 0],
    ], dtype = float) # yapf: disable
    data['demands'] = 2
    np.fill_diagonal(data['distance_matrix'], np.inf)
    matrix = MatrixColony(data['distance_matrix'], data['demands'])

    print(matrix.delivery_nodes)
    print(matrix.pickup_nodes)

    antcolony = AntColony(matrix)
    print(antcolony.run())

```

Annexe 2 : Résultats obtenu lors de la première étude :

Beta1	Beta2	Beta3	Beta4	Beta5
9386.8	8735.871794871795	8375.086206896553	8146.584415584415	7978.46875
8730.55	8312.384615384615	7992.258620689655	7819.376623376624	7702.270833333333
8508.95	8075.820512820513	7780.896551724138	7639.649350649351	7558.104166666667
8322.25	7936.282051282052	7669.9655172413795	7536.766233766234	7463.34375
8111.9	7796.7692307692305	7550.551724137931	7421.441558441558	7359.21875
7997.0	7650.410256410257	7448.913793103448	7342.298701298701	7294.885416666667
7941.5	7578.410256410257	7381.4655172413795	7275.7532467532465	7218.9375
7793.35	7500.282051282052	7328.931034482759	7226.272727272727	7178.989583333333
7671.75	7403.641025641025	7253.0344827586205	7152.584415584415	7119.885416666667
7671.75	7403.641025641025	7253.0344827586205	7150.116883116883	7117.90625
7584.5	7358.897435897436	7221.517241379311	7124.558441558442	7084.1875
7546.45	7325.820512820513	7186.689655172414	7098.324675324675	7063.145833333333
7546.45	7318.897435897436	7181.362068965517	7084.480519480519	7052.041666666667
7492.2	7284.7692307692305	7158.258620689655	7067.077922077922	7031.479166666667
7463.75	7245.0	7130.5344827586205	7046.194805194805	7014.729166666667
7462.5	7244.358974358975	7130.103448275862	7045.87012987013	7013.114583333333
7439.75	7200.179487179487	7100.396551724138	7015.2467532467535	6988.552083333333
7385.2	7172.205128205128	7081.586206896552	7001.077922077922	6977.177083333333
7365.05	7149.153846153846	7055.137931034483	6975.2467532467535	6955.364583333333
7347.7	7139.0	7047.4655172413795	6969.467532467533	6950.729166666667
7326.2	7127.974358974359	7040.051724137931	6963.883116883117	6946.25
7308.35	7118.820512820513	7033.862068965517	6959.220779220779	6942.510416666667
7259.55	7093.794871794872	7017.0344827586205	6946.545454545455	6932.34375
7241.75	7084.666666666667	7010.896551724138	6941.922077922078	6928.635416666667
7235.25	7073.076923076923	7003.103448275862	6936.051948051948	6923.927083333333
7212.05	7058.4358974358975	6993.258620689655	6928.636363636364	6917.979166666667
7182.8	7043.4358974358975	6983.172413793103	6921.038961038961	6911.885416666667
7097.55	6998.25641025641	6941.4655172413795	6889.623376623376	6886.6875
7097.5	6998.205128205128	6941.431034482759	6889.597402597403	6886.666666666667
7097.5	6998.205128205128	6941.431034482759	6889.597402597403	6886.666666666667

Annexe 3 : résultats obtenus lors de la deuxième étude :

Pour la première simulation :

0	8804.0	7497.0	7334.0	10029.0	9069.0	8445.0
1	8019.0	8029.0	8202.0	8333.0	8637.0	7984.0
2	9859.0	8615.0	7581.0	7440.0	8213.0	8747.0
3	8765.0	8202.0	8533.0	8752.0	7396.0	8538.0
4	8644.0	8761.0	7876.0	7020.0	9018.0	9061.0
5	7332.0	8214.0	8320.0	8006.0	7847.0	7475.0
6	8406.0	7876.0	7888.0	8173.0	8457.0	7594.0
7	7935.0	7750.0	6698.0	8791.0	8236.0	8546.0
8	8205.0	8400.0	7669.0	8747.0	7681.0	7915.0
9	6698.0	8635.0	7993.0	8052.0	7511.0	8202.0
10	8267.0	8035.0	7154.0	8009.0	7570.0	8962.0
11	8053.0	7837.0	7359.0	7909.0	7954.0	8052.0
12	8317.0	7333.0	7857.0	8429.0	8722.0	7847.0
13	8367.0	8062.0	6698.0	8438.0	6699.0	9162.0
14	7194.0	7851.0	7635.0	9314.0	7194.0	7914.0
15	7914.0	8594.0	9043.0	8739.0	7154.0	7974.0
16	8966.0	8658.0	8159.0	7877.0	7914.0	8380.0
17	9318.0	7498.0	7967.0	8203.0	8466.0	8407.0
18	7332.0	8711.0	8348.0	7440.0	8786.0	8510.0
19	7964.0	7935.0	8581.0	8548.0	9148.0	7440.0
20	7814.0	7593.0	7914.0	6698.0	8383.0	7672.0
21	8865.0	8459.0	8797.0	8291.0	8681.0	7382.0
22	8273.0	8019.0	7194.0	8202.0	7774.0	7766.0
23	7681.0	8269.0	7570.0	6698.0	7497.0	8135.0
24	7847.0	7475.0	7676.0	8102.0	6699.0	8757.0
25	7909.0	7857.0	8728.0	8202.0	8276.0	7993.0
26	9058.0	8084.0	8584.0	8019.0	8222.0	7334.0
27	7656.0	8471.0	8019.0	8302.0	8119.0	7428.0
28	9161.0	7909.0	8852.0	7689.0	7332.0	7636.0
29	7471.0	8435.0	8720.0	7773.0	7672.0	7914.0
30	8669.0	9070.0	8067.0	7594.0	8282.0	7723.0
31	8701.0	7594.0	8246.0	7636.0	8552.0	7909.0
32	7655.0	8397.0	8591.0	7593.0	7758.0	8022.0
33	8456.0	8205.0	7713.0	8270.0	7857.0	7914.0
34	9337.0	7750.0	7334.0	8985.0	7847.0	7924.0
35	8061.0	8535.0	8953.0	7334.0	7649.0	7942.0
36	7676.0	8228.0	8447.0	8632.0	7717.0	8584.0
37	9288.0	7914.0	8263.0	8288.0	7636.0	8045.0
38	7711.0	7484.0	7193.0	7675.0	8997.0	8057.0
39	8860.0	7928.0	8850.0	7154.0	7997.0	7840.0
40	8606.0	8149.0	8340.0	8735.0	7020.0	8110.0
41	7833.0	9805.0	7580.0	9018.0	7915.0	8173.0
42	8466.0	7795.0	8009.0	7967.0	7194.0	8341.0
43	7994.0	8274.0	7484.0	7672.0	8291.0	8494.0
44	7670.0	7787.0	10138.0	7994.0	7594.0	8424.0
45	7557.0	9057.0	7732.0	7497.0	9559.0	8731.0
46	8813.0	7750.0	8466.0	8205.0	8754.0	7713.0
47	8087.0	7681.0	8469.0	7993.0	7793.0	8495.0
48	7814.0	8421.0	8214.0	8567.0	7334.0	8742.0
49	8029.0	9248.0	8045.0	8214.0	8348.0	8274.0

Pour la seconde simulation :

0	7511.0	6698.0	7672.0	7334.0	7909.0	7153.0	7276.0	7711.0	7497.0	7332.0
1	7669.0	7440.0	7593.0	7914.0	7857.0	7570.0	6699.0	7711.0	7935.0	7711.0
2	7275.0	8243.0	7194.0	6699.0	7154.0	6699.0	7675.0	6698.0	7561.0	7276.0
3	7428.0	7594.0	6698.0	7676.0	7473.0	7276.0	7670.0	7724.0	7193.0	7711.0
4	8507.0	7847.0	7515.0	7396.0	7333.0	6699.0	7440.0	7840.0	7154.0	7792.0
5	7276.0	7464.0	7439.0	7674.0	7334.0	7194.0	7669.0	6699.0	7334.0	7669.0
6	7440.0	8424.0	7792.0	7359.0	6699.0	7592.0	7194.0	7561.0	7473.0	7840.0
7	7472.0	7594.0	7276.0	7396.0	7382.0	7464.0	6698.0	7193.0	7194.0	7332.0
8	7553.0	7914.0	7332.0	7439.0	7909.0	7193.0	7594.0	6699.0	7194.0	7333.0
9	8214.0	7681.0	7594.0	7957.0	7681.0	7334.0	7439.0	7194.0	7847.0	7557.0
10	7713.0	7554.0	6699.0	6699.0	7275.0	6699.0	7194.0	7154.0	7428.0	7464.0
11	7194.0	7847.0	6699.0	7554.0	7472.0	7020.0	7332.0	7787.0	7193.0	6698.0
12	7440.0	7333.0	7275.0	6699.0	7594.0	6698.0	7333.0	6698.0	6698.0	7471.0
13	7713.0	7464.0	7713.0	7557.0	6699.0	7359.0	7473.0	6698.0	7594.0	7635.0
14	8029.0	6698.0	7711.0	7154.0	7557.0	6699.0	7674.0	7153.0	7471.0	7580.0
15	7708.0	7593.0	7396.0	7554.0	7276.0	7726.0	7695.0	7635.0	7674.0	7333.0
16	7020.0	7689.0	7498.0	7020.0	6698.0	6699.0	7511.0	6699.0	7924.0	6699.0
17	7581.0	7464.0	7153.0	6698.0	7681.0	7847.0	7153.0	7332.0	7511.0	7674.0
18	7497.0	7557.0	7773.0	6698.0	7382.0	7928.0	6699.0	7847.0	7334.0	7334.0
19	7276.0	7440.0	7669.0	7847.0	6698.0	7935.0	7332.0	7636.0	7993.0	7498.0
20	8084.0	7333.0	7723.0	7581.0	7642.0	6698.0	7471.0	7570.0	7472.0	7498.0
21	7154.0	7935.0	7511.0	7428.0	7332.0	7275.0	6699.0	7773.0	7382.0	7428.0
22	7939.0	7724.0	7708.0	7675.0	7275.0	7967.0	7511.0	7332.0	7153.0	7511.0
23	7383.0	8057.0	7439.0	7439.0	6699.0	6698.0	7334.0	6699.0	7382.0	7333.0
24	7473.0	8228.0	7194.0	7553.0	7396.0	7439.0	7193.0	6699.0	7708.0	7382.0
25	7511.0	7484.0	6699.0	7332.0	7333.0	7275.0	7154.0	7382.0	7334.0	6698.0
26	7655.0	7847.0	7935.0	7334.0	7472.0	7511.0	7594.0	7383.0	7594.0	7382.0
27	7713.0	7592.0	6698.0	7994.0	7440.0	7580.0	7497.0	6698.0	7676.0	7429.0
28	7464.0	7332.0	7475.0	7334.0	7909.0	7594.0	7439.0	7154.0	7594.0	8002.0
29	8029.0	7594.0	7428.0	7902.0	7275.0	7332.0	7473.0	7334.0	7471.0	7428.0
30	7553.0	8341.0	7847.0	7334.0	7382.0	7681.0	6698.0	6698.0	7914.0	8052.0
31	7636.0	7561.0	7847.0	7592.0	7332.0	7333.0	7428.0	7276.0	7276.0	7732.0
32	7153.0	8276.0	7635.0	7924.0	7383.0	7464.0	7675.0	7154.0	6698.0	7464.0
33	7676.0	7497.0	7396.0	7707.0	7332.0	7594.0	8427.0	7153.0	7153.0	7764.0
34	8233.0	7471.0	7333.0	8057.0	7773.0	7792.0	7475.0	7471.0	7471.0	7594.0
35	7888.0	8274.0	7153.0	7581.0	7332.0	7440.0	7592.0	7498.0	7153.0	7153.0
36	7334.0	7332.0	7194.0	7840.0	7276.0	7642.0	7847.0	7471.0	7594.0	7440.0
37	6698.0	7593.0	6699.0	7764.0	7194.0	7472.0	7276.0	7909.0	7669.0	6699.0
38	7914.0	7383.0	7689.0	7333.0	7020.0	7332.0	7580.0	7766.0	7511.0	7382.0
39	7464.0	7707.0	7334.0	7153.0	6698.0	7959.0	7275.0	7275.0	7383.0	7154.0
40	7471.0	7858.0	7968.0	6698.0	7726.0	8276.0	7511.0	7332.0	7594.0	7561.0
41	7807.0	7553.0	7511.0	7695.0	7440.0	7333.0	7333.0	7672.0	7332.0	6699.0
42	7154.0	7674.0	7472.0	7333.0	7333.0	7334.0	7924.0	7484.0	7439.0	7471.0
43	8397.0	7993.0	7773.0	7275.0	6699.0	7674.0	7902.0	6698.0	7649.0	7594.0
44	7814.0	7333.0	7383.0	7581.0	7985.0	7497.0	7396.0	7154.0	7333.0	7670.0
45	7963.0	8203.0	7194.0	7840.0	6698.0	7511.0	7440.0	7276.0	7914.0	7472.0
46	7383.0	7020.0	7580.0	7704.0	7684.0	7332.0	6698.0	7914.0	7676.0	7511.0
47	8196.0	7153.0	7333.0	7511.0	7909.0	7334.0	7553.0	7840.0	7153.0	7333.0
48	7924.0	7914.0	7964.0	6698.0	7795.0	7711.0	6699.0	7847.0	7332.0	7153.0
49	8009.0	8045.0	7473.0	7154.0	6698.0	7707.0	7774.0	7580.0	7726.0	7334.0