

Planification De l'approvisionnement Des Stations D'essence

Planning of the supply of gas stations

Abdali Abdekrim *

(Laboratoire de la mondialisation et
politique économique),
Université d'alger3, algérie
abdali.abdekrim@univ-alger3.dz

bouzara Laid

(Laboratoire de la mondialisation
et politique économique),
Université d'alger3, algérie
dzbouzara.laid@univ-alger3.dz

Date de réception : 09/04/2022 ;

Date d'acceptation : 28/05/2022

Résumé :

L'objectif de ce travail est d'identifier les itinéraires que suivent les camions citernes pour réduire le temps et la distance parcourus pour approvisionner 24 stations-service à Alger. Ce problème est connu dans la littérature sous le nom de problème de routage de véhicule avec fenêtres temporelles (VRPTW). Dans ce contexte, nous avons évalué 3 modèles de programmation mathématique en nombre entier, où les résultats obtenus ont montré que le modèle d'agrégation d'objectifs est le plus robuste en termes de critères de durée et de distance parcourue.

Mots clés : Programmation mathématique, théorie des graphes, problème de tournée de véhicule, problème de (VRPTW).

Classification JEL: C610.

Abstract:

The objective of this work is to identify the routes followed by tank trucks to reduce the time and distance traveled to supply 24 service stations in Algiers. This problem is known in the literature as the Vehicle Routing Problem with Time Windows (VRPTW). In this context, we evaluated 3 integer mathematical programming models, where the results obtained showed that the goal aggregation model is the most robust in terms of duration and distance traveled criteria.

Keywords: Mathematical programming, graph theory, vehicle routing problem, (VRPTW) problem.

Jel Classification Codes : C610

1. Introduction:

* L'auteur correspondant Abdali Abdekrim : abdali.abdekrim@univ-alger3.dz

Notre travail consiste à traiter le problème **de tournées de véhicules avec fenêtre temporelle** pour tenter d'optimiser et planifier l'approvisionnement des stations d'essence à Alger en respectant la contrainte de fenêtre temporelle, pour un nombre de **24 stations** à partir d'un seul dépôt de **Kherroubi**. Dans ce contexte, nous posons la problématique suivante :

« Quels itinéraires les camions doivent-ils suivre pour minimiser la durée et la distance parcourus afin d'approvisionner les stations-service ? »

Pour répondre à cette problématique Nous évaluerons 3 modèles de programmation linéaire avec des nombres entiers. Le but est du minimiser la distance et la durée de tourner de chaque camions. En précisant le nombre de camions à utiliser, l'itinéraire et la durée requise pour chaque camion.

L'ensemble des contraintes reste le même pour les 3 modèles, la seule différence sera au niveau de la fonction objectif. Pour le 1er modèle la fonction objectif vise à minimiser la somme totale de temps de parcours, pour le 2eme modèle la fonction objectif vise à minimiser la distance totale parcourue par les véhicules et pour le 3eme modèle la fonction objectif vise à minimiser les deux précédentes fonctions simultanément en une seule fonction d'agrégation.

Les hypothèses :

Les principales hypothèses liées à l'étude sont les suivantes :

- Cette approche ne s'applique que pour un planning à court termes avec un nombre limité de stations et un seul dépôt avec un seul type de véhicule.
- L'approvisionnement s'effectuera par voie routière à partir de dépôts de Kherroubi.
- Le planning est pour un seul type de carburant (****gaz oil****).
- La demande des stations est supposée connue et déterministe.
- Chaque station doit être visitée une fois par un seul véhicule sur la période

Le travail est divisé en deux parties. Dans la première partie, une brève présentation de la littérature sur le problème du VRP et de ses variantes sera donnée. Dans la deuxième partie, nous appliquerons un algorithme **branch&bound** pour résoudre les trois modèles impliqués. Cela permettra de planifier un approvisionnement à court terme de stations-service répondant

aux critères de la fenêtre temporelle, du nombre de véhicules, de la distance parcourue et de la durée du trajet.

2. La partie théorique

2.1 Lecteur dans la littérature scientifique

- ❖ (G.B.Dantzig & Ramser, 1959) ont publié pour la première fois un article sur le problème de tournée de véhicule (VRP) en 1959 (RACHID, CHERIF, BLOCH, & CHATONNAY, 2008). La formule développée par Dantzig consiste à préciser le chemin qu'une voiture doit suivre pour visiter un groupe de villes et revenir au même point de départ. Depuis, plusieurs travaux ont été réalisés, justifiés par l'impact du transport sur le coût total de la marchandise. Parmi ceux trouvés:
- ❖ (C.Krishnaraj; A.Anand Jayakumar; S.Deepa Shri, 2015) article publié en 2015 sous le titre «**Solving Supply Chain Network Optimization Models Using LINGO** ». Les auteurs ont présenté trois modèles différents pour résoudre le problème de transport classique, les résultats ont montré que l'approche multi-objective est la plus robuste pour confronter ce type de problème.
- ❖ (Mohan, Chiriki; Lee, Yooneun; Prabhu, Vittaldas V., 2017). Cet article présente trois approches différentes pour modéliser et résoudre le problème classique de localisation de transport, y compris la programmation linéaire traditionnelle en nombres entiers mixtes de minimisation des coûts. Ils ont montré que la modélisation de ces problèmes comme un programme mixte de maximisation du profit en nombres entiers permet plutôt la relaxation d'un ensemble de contraintes.
- ❖ (Han, Ming; Ya-Bin Wang, 2020). cet article propose une idée de solution en deux étapes d'ajustement dynamique de la trajectoire de pré-planification pour résoudre le **VRPTW**. Dans l'étape initiale de la planification du chemin, basée sur des facteurs de guerre, cet article construit un modèle d'optimisation des itinéraires de véhicules de distribution en temps de guerre avec des fenêtres temporelles dans un environnement complexe, et conçoit un algorithme génétique et un algorithme auto-adaptatif.

2.2 Classification de -vehicle routing problem- (VRP) :

Il existe de nombreuses variantes du problème VRP qui ont été étudiées depuis 1950. Cette variante est relativement liée aux variations des objectifs, des contraintes et des variables de décision. Le **VRP** est généralement modalisé sous forme d'un graphe non orienté. Le but de cette partie du document est de donner un aperçu général sur les différents types de problèmes de **VRP**, pour plus de détail voir (RACHID, CHERIF, BLOCH,

& CHATONNAY, 2008), (Dounya Taqa allah, Jacques Renaud, & Faye F. Bector)

Définition :

Le problème de tournée de véhicule (**VRP**) est une extension de problème classique de voyageurs de commerce (Traveling Salesman Problem –**TSP**-), introduit pour la première fois par **Dantzig et al** en 1959. L'idée de **VRP** est de déterminer la tournée d'un véhicule qui doit desservir un ensemble des clients. Chaque tournée commence et se termine au dépôt (Ismail, Legras, & Coppin, 2011, p. 6). Le but est de minimiser le coût total de transport.

Le problème de capacity vehicle problem (c-vrp) :

Le principe du (C-VRP) est de déterminer les itinéraires d'un véhicule tout en veillant à la satisfaction de chaque client. Dans ce cas la capacité du véhicule est illimitée, la visite commence et se termine au dépôt unique (B.M.BAKER & M.A.AYECHW, 2003, pp. 787- 800).

Les Caractéristiques de **C-VRP** :

Données :

- la demande des clients connus est déterministe.
- Le nombre de clients est limité.
- Un point de départ et de retour unique.

Objectif :

- Minimiser le coût total de transport.

Résultats :

- La détermination des tournées de véhicule.
- La satisfaction de la demande.
- Chaque client doit être visité une seule fois.

Le problème de (VRPTW) vehicle routing problem with time windows:

Dans le cas de (VRPTW) chaque client doit être servi dans une fenêtre temporelle, il s'agit d'un intervalle de temps pendant lequel son service et chaque client doivent être visités une et une seule fois (Mais haj rachid, 2010, pp. 13-14).

Il existe deux types de problème (**VRPTW**), le problème avec une fenêtre temporelle dure et le problème avec une fenêtre temporelle souple.

a. le (VRPTW) avec fenêtre dure :

Dans ce cas, si le véhicule arrive avant la fenêtre horaire, il doit attendre l'ouverture de la station, et s'il arrive trop tard il ne pourra pas effectuer la livraison.

b. le (VRPTW) avec fenêtre souple :

Dans ce cas le véhicule peut servir le client en dehors de la fenêtre horaire mais avec une certaine fonction de pénalité.

Les caractéristiques de problème de **(VRPTW)** :

Données :

- la demande des clients connus est déterministe.
- La fenêtre temporelle souple/dure.
- La flotte est supposée homogène
- Un point de départ et de retour unique.
- Le nombre de véhicules peut être fixe d'avance ou être considéré comme l'une des variables de décision.

Objectif :

- Minimiser le coût total de transport.
- Minimiser le nombre de véhicules utilisés.
- Minimiser la distance à parcourir.

Résultats :

- La détermination des tournées de véhicule.
- La détermination de nombre des véhicules nécessaires.
- La satisfaction de la demande.
- Chaque client doit être visité une seule fois.
- La détermination de la distance à parcourir.
- La détermination de la durée de parcours

Le problème de (MD-VRP) multi-depot vehicle routing problem :

Dans ce type de problèmes, les véhicules disponibles sont situés dans plusieurs dépôts, chaque tournée doit être commencée et terminée au même dépôt de départ, et chaque client ne doit être desservi qu'une seule fois par un seul véhicule (A.MAINGOZZI, 2005, pp. 347- 350).

Les caractéristiques de problème de **(MD-VRP)**:

Données :

- la demande des clients connus sur l'horizon de planification connus est déterministe.

- La flotte est supposée homogène
- Plusieurs dépôts.
- Le nombre de véhicules est fixe.

Objectif :

- Minimiser le coût total de transport.

Résultats :

- La détermination des tournées de véhicule.
- La détermination du nombre des véhicules nécessaires.
- La détermination de la distance à parcourir.
- La détermination de la durée de parcours.

Le problème de (S-VRP) stochastic vehicle routing problem :

La spécification pour le problème de **(S-VRP) stochastic vehicle routing problem**, c'est qu'un ou plusieurs paramètres du problème (demande, durée, nombre de client...etc.) sont aléatoires.

Les caractéristiques du problème de **S-VRP** :

Données :

- la demande de client est aléatoire (non déterministe).
- La flotte est supposée homogène
- Un ou plusieurs dépôts.
- La durée de trajet est aléatoire.

Objectif :

- Minimiser le coût total de transport.

Résultats :

- La détermination des tournées de véhicule.
- La détermination du nombre des véhicules nécessaires.
- La détermination de la distance à parcourir.
- La détermination de la durée de parcours

2.2 Méthodes de résolution de problème de tournée de véhicule:

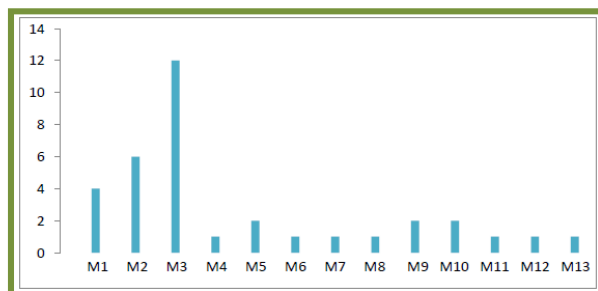
Pour résoudre un problème VRP formulé sous forme d'une programmation linéaire en nombres entiers, il existe deux types de méthodes, les méthodes exactes et les méthodes approchées. Selon (G.LAPORTE & Y.NOBERT, 1987, pp. 147- 184), les trois principales méthodes exactes pour résoudre un problème de **VRP** sont ; **Branch&bound, branch&cut et la programmation dynamique**, ces méthodes donnent une solution dite optimale dans un temps de calcul lié à la taille du problème (nombre et type

de variable de décision, nombre d'objectifs, données, etc...). L'inconvénient de ce type de méthodes est qu'elles sont limitées par la taille du problème, elles ne peuvent être appliquées qu'à court terme avec relativement peu de clients.

(T.K.RALPHS, 2003, pp. 607- 629) a proposé une solution avec un algorithme de **branch& cut** pour résoudre un problème de **VRP** contenant 100 clients.

Le deuxième type de méthodes appelées méthodes approchées, elles ne donnent pas une solution optimale mais un ensemble de solutions approchées appelées front de Pareto. Ces méthodes sont conçues pour résoudre de gros problèmes dans un temps de calcul raisonnable, Parmi ces méthodes, on trouve: recherche tabou, algorithme génétique, colonne formée... etc. Pour plus de détails sur les méthodes approchées voir (PATRICK SIARY&AL, 2003) ; (Lobato & Valder Steffen Jr, 2017) ; (Maniezzo, Stutzle, & Stefan Voß, 2009).

- Figure 1 : Diagramme Pareto de la fréquence d'utilisation des méthodes traitées



- Source: (Bessid, Alaeddine, & benabdelhafid, 2018, p. 4)
- ➤ M1: Programmation Linéaire (PL)
- ➤ M2: Generalized Linear Mixed Models (GLMM)
- ➤ M3: Algorithmes génétiques (heuristique)
- ➤ M4: Optimisation Intégrée (OI)
- ➤ M5: Modélisation et Simulation des Réseaux Intermodaux (MSRI)
- ➤ M6: Ant Colony Optimization (ACO)
- ➤ M7: Mixed Integer Linear Programming (MILP)
- ➤ M8: Nonlinear Fixed Charge Transportation Problem (NCFTP)

- ➤ M9: Multi-echelon (échelons multiples) (EM)
- ➤ M10: Méthode d'aide à la décision(MAD)
- ➤ M11: Hybridation d'une méta-heuristique
- ➤ M12: Une méthode d'estimation(ME)
- ➤ M13: Activity-based costing. (ABC)

3. Optimisation et planification d'approvisionnement des stations d'essence à Alger avec fenêtre temporelle - étude de cas NAFTAL el Harrach

Dans cette deuxième partie du document, nous voulons construire trois modèles de programmation linéaire mixte en nombres entiers. L'objectif est d'évaluer les 3 modèles pour optimiser et planifier à court terme l'approvisionnement d'un ensemble de stations-service à Alger, pour un nombre de **24 stations**. Supposant que la demande de chaque station est connue et déterministe sur la période, avec une fenêtre temporelle fixe pour toutes les stations, la distribution est réalisée à partir d'un seul dépôt à **Kherroubi**.

❖ Hypothèses relatives aux problèmes étudié:

Nous intéressons au problème de tournée de véhicules avec fenêtre temporelle. D'abord on va définir l'ensemble des hypothèses qui servent à simplifier le problème et déterminent les limites de ce travail.

- Cette approche ne s'applique que pour un planning à court termes avec un nombre limité de stations et un seul dépôt avec un seul type de véhicule (homogène).
- L'approvisionnement s'effectuera par voie routière à partir de dépôts de Kherroubi.
- Le planning est pour un seul type de carburant (**gaz oil**).

❖ Hypothèses relatives aux stations :

- La demande des stations est supposée connue et déterministe.
- L'approvisionnement s'effectue dans une fenêtre temporelle (la même pour toutes les stations).
- Chaque station doit être visitée une fois par un seul véhicule sur la période.
- La quantité demandée de chaque station doit être inférieure ou égale à la capacité de véhicules.

❖ **Hypothèses relatives aux véhicules :**

- La durée de tournées ne doit pas dépasser la durée maximale de travail.
- Pas de heurts supplémentaires.
- Les véhicules sont homogènes (mêmes : capacité, vitesse,...etc.)
- Les véhicules transportent un seul type de produit.
- Le nombre de véhicules est considéré comme une variable à déterminer.

❖ **Hypothèses relatives au dépôt :**

- Un seul dépôt.
- La quantité du carburant disponible est supposée suffisante.
- La durée de chargement de véhicules est la même pour toute la flotte.

La modalisation :

1. Fonction objectif :

Pour réaliser une bonne planification nous évaluerons 3 modèles, visant deux objectifs principaux, la minimisation de la distance parcourue et la minimisation du temps de trajet. Pour les 3 modèles, toutes les contraintes restent les mêmes à quelques détails près, la différence sera au niveau de la fonction objectif comme :

• **Modèle 1 : fonction objectif 1 :**

Minimiser la durée totale de parcours.

• **Modèle 2 : fonction objectif 2 :**

Minimiser la somme des distances totales parcourues.

• **Modèle 3 : fonction objectif 3 :**

Minimiser la distance et la durée de parcours simultanément.

• **Formulation mathématique :**

- **fonction objectif 1 :** $\min f1 = \sum T_ARRIVER(i)$.
- **fonction objectif 2 :** $\min f2 = \sum \sum DIST(i, j) * Z(i, j)$
- **fonction objectif 3 :** $\min f3 = \sum \sum DIST(i, j) * Z(i, j) + 0.001 * \sum T_ARRIVER(i)$

2. Variables de décision :

- **Z(i,j)** « variable binaire » = 1 si l'arc (i, j) est traversé par le véhicule, sinon 0.
- **T_ARRIVER(i)** = le temps d'arrivée à la station i.
- **quantite_cum(i)** = quantité de produits dans le véhicule après le passage à la station i.

« Le rôle de cette variable est de calculer le volume de carburant restant dans le véhicule après le passage de station (i) ».

- **Distance_cum(i)** = la somme de distance cumulée jusqu'à la station i.

3. Les données :

Les données sont fournies par **NAFTAL** :

- La vitesse de véhicule= 70km/1h = 1.2km/1min
- Le nombre de véhicules disponibles (supposé suffisant) =99999
- Le nombre de stations pour une période = 24 stations.
- Capacité de chaque véhicule : $V_CAP=30 \text{ m}^3$
- Les distances entre les stations : $\text{dist}(i,j)$
- Quantité demandée de chaque station = $Q(i)$

Tableau (1) : les commandes de 24 stations sur la période.

Station	1	2	3	4	5	6	7	8	9	10	11	12
Q(i) Commande (m^3)	15	14	20	14	30	60	13	13	15	15	8	27
Station	13	14	15	16	17	18	19	20	21	22	23	24
Q(i) Commande(m^3)	6	13	15	25	25	30	27	15	16	30	30	30

Source : NAFTAL

- La durée de service sur chaque station i $T_SERVICE(i)$

Tableau(2) : La durée de service estimée sur chaque station

Station	1	2	3	4	5	6	7	8	9	10	11	12
T_service (min)	30	28	40	28	60	12	26	26	30	30	16	54
Station	13	14	15	16	17	18	19	20	21	22	23	24
T_service (min)	12	26	30	50	50	60	54	30	32	60	60	60

Source : NAFTAL

- La fenêtre temporelle : $[t_debut=19.00h ; t_fin 7.00 \text{ pm}]$
- Durée max de service = la distance de la fenêtre : $12h = 7200\text{min}$.
donc on suppose la fenêtre $[t_debut=0 ; t_fin 7200 \text{ min}]$.

4. Les contraintes :

Les contraintes représentent les caractéristiques de problème :

- La contrainte de fenêtre temporelle.
- La contrainte de demande.
- La contrainte de distance maximale parcourue par chaque véhicule.
- La contrainte de capacité de véhicule.
- Chaque station doit desservir une et une seule fois.
- Les contraintes de binarité sur les variables de décision x_{ijk} .
- Assurer que chaque tournée commence et se termine au dépôt

Nous avons codé les contraintes directement sur le logiciel de LINGO_SOFTWARE (voir l'annexe1).

La résolution de modèles :

Pour résoudre les 3 modèles nous utiliserons la technique branch&bound, la formulation précédente est codée sur le logiciel lingo soft 18.01 (voir annexe 1), à l'aide du guide (LINGO, 2020).

Tableau 2 : la résolution des trois modèles avec lingo software 18.0

Critères Modèles	Fonction objectif	Nombre des véhicules	Total distance (km)	Total durée (min)
1	Min F1 : la durée de parcours	24	1350	810
2	Min F2 : la distance parcourue	16	1018	1091
3	Min F3 : la durée et la distance simultanément	16	1018	1005

Source : par les auteurs à l'aide de lingo software 18.0

Les résultats du tableau (2) montrent que si l'on minimise seulement le temps de parcours (**MinF1**) le nombre de véhicules sera de **24 camions** avec une distance totale parcourue de **1350 (Km)** mais avec un temps total minimum de **810 min**. Pour la deuxième fonction (**MinF2**) qui minimise la distance seule, le nombre de véhicules sera de **16 camions** avec une distance totale de **1018 (KM)** moins que pour le modèle de (**Min F1**) mais avec un temps total plus long de (**1091 min**). Dans le cas de la minimisation simultanée des deux fonctions (**Min F3**), le nombre de véhicules qui sera utilisés est de **16 camions** avec une distance totale de déplacement de **1018 km** et une durée totale de **1005 min**.

Ce que l'on peut déduire du tableau(2) c'est que le modèle (Min F3) donne une solution de compromis entre les différents critères, cela nous pousse à le choisir pour planifier l'approvisionnement des stations.

Résultat final :

Notre objectif est de construire un planning à court terme pour optimiser l'approvisionnement des stations-service à Alger. Après l'évaluation des trois modèles, les résultats du tableau 2 montrent que le modèle d'agrégation (modèle 3) donne les meilleurs résultats en ce qui concerne les critères de distance, de durée et de nombre de véhicules utilisés.

Nous adopterons donc le troisième modèle. Le résultat final concerne les trajets, l'heure d'arrivée aux gares, la distance totale, le temps total et le nombre de véhicules utilisés sont indiqués dans le tableau suivant :

Tableau (3) : la planification de l'approvisionnement des stations d'essence.

Véhicules	Point de départ	Point d'arriver	durée cumulée (min)
Véhicule 1	DEPOT	RUE_N5	14.4
	RUE_N5	ROUIBA	52.0
	ROUIBA	DEPOT	115.2
Véhicule 2	DEPOT	BABA_HACENE	36.0
	BABA_HACENE	DEPOT	132.0
Véhicule 3	DEPOT	STATION_AUTOROUTE	24.0
	STATION_AUTOROUTE	OUEL_MOUSSA	62.0
	OUEL_MOUSSA	DEPOT	126.0
Véhicule 4	DEPOT	HEURAOUA	31.2
	HEURAOUA	REGHAIA	67.2
	REGHAIA	DEPOT	141.6
Véhicule 5	DEPOT	BOURDJ_EL_BAHRI	25.2
	BOURDJ_EL_BAHRI	DERGANA	61.2
	DERGANA	DEPOT	114.8
Véhicule 6	DEPOT	AIN_TAYA	33.6
	AIN_TAYA	OULED_MOUSSA2	73.6
	OULED_MOUSSA2	DEPOT	149.6
Véhicule 7	DEPOT	AIN_NAADJA	12.0
	AIN_NAADJA	DEPOT	78.0
Véhicule 8	DEPOT	EL_HARRACH	8.4
	EL_HARRACH	MEFTAH	46.8
	MEFTAH	LARBAA	70.8
	LARBAA	DEPOT	138.8
Véhicule 9	DEPOT	TESSALA	36.0
	TESSALA	DEPOT	122.0
Véhicule 10	DEPOT	TESSALA2	37.2
	TESSALA2	DEPOT	124.4
Véhicule 11	DEPOT	HAMMEDI	46.8
	HAMMEDI	DEPOT	153.6
Véhicule 12	DEPOT	HAMMEDI2	48.0
	HAMMEDI2	DEPOT	150.0
Véhicule 13	DEPOT	BARAKI	24.0
	BARAKI	SIDI_MOUSSA	62.4
	SIDI_MOUSSA	DEPOT	133.2

Véhicule 14	DEPOT	MOUILHA	54.0
	MOUILHA	DEPOT	168.0
Véhicule 15	DEPOT	KHEMIS_EL_KHECHNA	36.0
	KHEMIS_EL_KHECHNA	DEPOT	132.0
Véhicule 16	DEPOT	BOUDOUAOU	43.2
	BOUDOUAOU	DEPOT	146.4

Source : les auteurs à l'aide de solveur **Lingo_software** 18.0

Conclusion :

L'optimisation des coûts de transport reste une priorité essentielle au quotidien quelle que soit l'entreprise, cette importance est justifiée par l'impact du transport, qui représente entre 10% et 20% du coût total des produits finis (Ismail, Legras, & Coppin, 2011, p. 5). Notre travail a consisté à optimiser et planifier l'approvisionnement des stations-service à partir d'un seul dépôt, sous contrainte de fenêtre temporelle. La planification se fait en définissant les trajets de chaque véhicule, les heures d'arrivée aux stations, et en déterminant le nombre de véhicules utilisés. La programmation mathématique nous offre un outil d'aide à la décision efficace qui nous permet de simplifier et de formaliser ce type de problème.

Dans ce document nous avons évalué trois modèles de programmation linéaire déterministe en nombres entiers mixtes pour tenter de planifier l'approvisionnement de 24 stations-service à Alger à partir d'un seul dépôt de « **kharouba** » afin d'optimiser le coût de transport, tout en respectant la contrainte de fenêtre temporelle. Les résultats montrent que le modèle d'agrégation est le plus robuste dans notre étude de cas, car il prend en compte les différents objectifs simultanément. La résolution a été réalisée par la technique de branch & bound, les principaux résultats sont les suivants :

- Le nombre de véhicules nécessaires pour effectuer l'approvisionnement est **16**
- La durée totale de parcours est **1005min**
- La distance totale parcourue par les véhicules est **1018km**
- Quantité totale de gaz oil transportée par les véhicules est **506m³**.

Bibliographie

- A.MAINGOZZI. (2005). THE MULI-DEPOT PERIODIC VEHICLE ROUTING PROBLEM. In *abstraction, reformulation and approximation* (pp. 347- 350).
- B.M.BAKER, & M.A.AYECHHEW. (2003). a genetic algorithem for vehicule routing problem . *compters end operations research*, pp. 787-800.
- Bessid, S., Alaeddine, Z., & benabdelhafid, A. (2018). Étude de l'optimisation des coûts de la chaîne de distribution: cas des coûts de stockage et de transport.
- C.Krishnaraj; A.Anand Jayakumar; S.Deepa Shri. (2015). Solving Supply Chain Network Optimization Models Using LINGO. *International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 19*.
- Dounya Taqa allah, Jacques Renaud, & Faye F. Boctor. (n.d.). Le problème d'approvisionnement des station d'essence. *Journal Européen des Systèmes Automatisés Volume X – n°X/2000*.
- G.B.Dantzig, & Ramser, J. (1959). The truck dispatching problem. *Operations Research. Management Sciences*, pp. 80–91.
- G.LAPORTE, & Y.NOBERT. (1987). EXACT ALGORITHMES FOR THE VEHICLE ROUTING PROBLEM. *annals of discrete mathimatics*, pp. 147- 184.
- Han, Ming; Ya-Bin Wang. (2020). A Solution to VRPTW Based on Improved GA-AMMAS. *Journal of Physics: Conference Series*.
- Ismail, S. B., Legras, F., & Coppin, G. (2011). *Synthèse du problème de routage de véhicules*. Bretagne: Collection des rapports de recherche de Télécom Bretagne.
- LINGO, S. (2020). Retrieved 02 07, 2021, from lingo: WWW: <http://www.lingo.com>
- Lobato, F. S., & Valder Steffen Jr. (2017). *Multi-Objective Optimization Problems, Concepts and Self-Adaptive Parameters with Mathematical and Engineering Applications*. Switzerland: springer.
- Mais haj rachid. (2010). thèse doctorat spécialité informatique. *le problème de tournées de vehicules en planification industrielle: classification et comparaiso d'opérateurs évolutionnaires*. france.
- Maniezzo, V., Stutzle, T., & Stefan Voß. (2009). *Matheuristics, Hybridizing Metaheuristics and Mathematical Programming*. Germany: springer.
- Mohan, Chiriki; Lee, Yooneun; Prabhu, Vittaldas V. (2017). Goal Programming for Supply Chain Optimization with Insufficient Capacity. *Springer International Publishing, APMS 2017, Part II, IFIP AICT 514.*, pp. 73–80.
- PATRICK SIARY&AL. (2003). *METHAHERISTIQUE*. france: eyrolles.

- RACHID, M. H., CHERIF, W. R., BLOCH, C., & CHATONNAY, P.
(2008). CLASSIFICATION DE PROBLEMES DE TOURNEES DE
VEHICULES. *7e Conférence Internationale de MOdélisation et
SIMulation - MOSIM'08 - du 31 mars au 2 avril 2008 – Paris-
France ;« Modélisation, Optimisation et Simulation des Systèmes :
Communication, Coopération et Coordination.».*
- T.K.RALPHS. (2003). PARALLET BRANCH AND CUT FOR
CAPACITATED VEHICLE ROUTING. *parallet computing
archive*, p. 607.

ANNEXE 1 :

```

MIN = DISTOT+0.01*temps_de_parcoure ;
      DISTOT = @SUM( CXC( i, j): DIST( i, j) * Z( i, j));
temps_de_parcoure=@SUM( STATION( i) | i #NE# DEPOTI:
T_ARRIVER( i));
      @FOR( STATION( k):
! Pour que le véhicule ne continue pas à tourner dans un
cercle vicieux;
      Z( k, k) = 0;
      );
      @FOR( STATION( k) | k #NE# DEPOTI:
! vehicule doit entrer dans la station K à partir d'une
certaine station I, ensemble
leurs exigences ne peuvent pas dépasser la capacité des
véhicules, et ne peut pas faire un trop long voyage;
[NTR] @SUM( STATION( i) | i #NE# k #AND# ( i #EQ# DEPOTI
#OR#
      (( Q( i) + Q( k) #LE# V_CAP) #AND# ( DIST( DEPOTI, i) +
DIST( i, k) + DIST( k, DEPOTI) #LE# DIST_ARRIVERX))) :
      Z( i, k)) = 1;
! Le véhicule doit quitter K après le service à une certaine
station J ;
[XIT] @SUM( STATION( j) | j #NE# k #AND# ( j #EQ# DEPOTI
#OR#
      (( Q( j) + Q( k) #LE# V_CAP) #AND# ( DIST( DEPOTI, k) +
DIST( k, j) + DIST( j, DEPOTI) #LE# DIST_ARRIVERX))) : Z( k,
j)) = 1;
! QUANTITE_CUM(k) est au moins la quantité nécessaire a K,
mais ne peut pas dépasser la capacité du véhicule;
@BND( Q( k), QUANTITE_CUM( k), V_CAP);
! Si K suit I, alors peut borner QUANTITE_CUM( k) -
QUANTITE_CUM( i);
@FOR( STATION( i) | i #NE# k #AND# i #NE# DEPOTI:
[LCUM] QUANTITE_CUM( k) >= QUANTITE_CUM( I)
      + Q( k)* Z( i, k) ! Case: i
precedes k;

```

```

- Q( i)* Z( k, i) ! Case: k
precedes i;
+ ( Q( k) - V_CAP)*(1- Z( i, k)-
Z( k, i)) ;
);
! Si K est le 1er arrat, alors QUANTITE_CUM( k) = Q( k);
QUANTITE_CUM( k) <= V_CAP - ( V_CAP - Q( k)) *
Z( DEPOTI, k);
! Si K est le 1er arrat...;
QUANTITE_CUM( k) >= Q( k) + @SUM( STATION( i)| I #NE#
DEPOTI: Q( i) * Z( i, k)););
! Calculer la distance totale parcourue par le véhicule grâce
a k ;
@FOR( STATION( k):
[R_TD_1] DISTANCE_CUM( k) >= DIST( DEPOTI, k)* Z( DEPOTI,
k);
@FOR( STATION( i)| i #NE# DEPOTI:
[R_TD] DISTANCE_CUM( k) >= DISTANCE_CUM( i)
+ DIST( i, k)* Z( i, k) ! Case: i precedes k;
- DIST_ARRIVERX * ( 1 - Z(i, k))
); );

DISTANCE_CUM( DEPOTI) <= DIST_ARRIVERX;
! Make the X's binary;
@FOR( CXC( i, j): @BIN( Z( i, j)));
! Contraintes liées a la fenêtre de temps ;
@FOR( STATION( k)| k #NE# DEPOTI:
@FOR( STATION( i) | i #NE# k:
! Heure d'arrivée a K si l'arrêt précédent était I ;
[RTM] T_ARRIVER( k) >= T_ARRIVER(i)
+ ( T_SERVICE( i) + TMPM* DIST( i, k))* Z( i,
k)
- T_FIN( i)*(1- Z( i, k));
! Chaque arrivée dans la fenêtre [T_DEBUT, T_FIN]. Nous
sommes autorisés à attendre afin d'arriver au plus tôt a
T_DEBUT(k);
@BND( T_DEBUT( k), T_ARRIVER( k), T_FIN( k)); );
! Contrainte de temps de trajet maximum;
[XT] T_ARRIVER( k) + T_SERVICE( k) + TMPM* DIST( k,
DEPOTI)* Z( k, DEPOTI) <= T_ARRIVERX; );
! Contrainte max véhicules;
@SUM( STATION( j) | j #NE# DEPOTI: Z( DEPOTI, j)) <=
MXTRK;
! Le Nombre minimum véhicules nécessaires, fractionnaire et
arrondi;
VEHCLF = @SUM( STATION( I)| I #NE# DEPOTI: Q( I))/ V_CAP;
VEHCLR = @FLOOR( VEHCLF +.9999);
! Doit envoyer suffisamment de véhicules hors du dépôt ;

```

```

@SUM( STATION( j) | j #NE# DEPOTI: Z( DEPOTI, j)) >=
VEHCLR;
! Sous_tour 2 coupes, impossible d'aller i à j et j à i,
( Case Q(i) + Q(j) #GT# V_CAP already taken care of);
@FOR( STATION( i) | i #NE# DEPOTI:
@FOR( STATION( j) | j #GT# i #AND# Q( i) + Q( j)
#LE# V_CAP:
Z( i, j) + Z( j, i) <= 1; ); );
! Sous_tours de coupes taille 3, ( Cas Q(i) + Q(j) #GT# V_CAP
déjà pris en charge);
@FOR( STATION( i) | i #NE# DEPOTI:
@FOR( STATION( j) | j #NE# DEPOTI #AND# i #LT# j :
@FOR( STATION( k) | k #NE# DEPOTI #and# j #LT# k:
[TRIPL] Z( i, j) + Z( j, i) + Z( i, k) + Z( k, i) + Z( j, k) +
Z( k, j) <= 2;
); ); );
! Couper : Impossible d'aller de i à j si le trajet est trop
long ;
@FOR( STATION( i) | i #NE# DEPOTI:
@FOR( STATION( j) | i #NE# DEPOTI #AND# j #NE# i #AND#
( T_SERVICE( i) + T_SERVICE( j) +
TMPM*( DIST( DEPOTI, i) + DIST( i, j) + DIST( j, DEPOTI))
#GT# T_ARRIVERX):
Z( i, j) = 0;
); );
! Couper : (i, j, k) ne peut pas être sur le même trajet s'il
est trop long ;
@FOR( STATION( i) | i #NE# DEPOTI:
@FOR( STATION( j) | j #NE# DEPOTI #AND# i #LT# j:
@FOR( STATION( k) | k #NE# DEPOTI #AND# j #LT# k
#AND#
( T_SERVICE( i) + T_SERVICE( j) + T_SERVICE( j) +
TMPM* @SMIN(
DIST( DEPOTI, i) + DIST( i, j) + DIST( j, k)
+ DIST( k, DEPOTI),
DIST( DEPOTI, i) + DIST( i, k) + DIST( k, j)
+ DIST( j, DEPOTI),
DIST( DEPOTI, j) + DIST( j, i) + DIST( i, k)
+ DIST( k, DEPOTI),
DIST( DEPOTI, j) + DIST( j, k) + DIST( k, i)
+ DIST( i, DEPOTI),
DIST( DEPOTI, k) + DIST( k, i) + DIST( i, j)
+ DIST( j, DEPOTI),
DIST( DEPOTI, k) + DIST( k, j) + DIST( j, i)
+ DIST( i, DEPOTI)) #GT# T_ARRIVERX): Z( i, j) + Z( i, k) +
z( j, i) + z( j, k) + z( k, i) + z( k, j) <= 1;); ); );
ENDSUBMODE

```