



# THEME :

## Le problème de tournées de véhicules

Réalisé par :

- ALKADMIRI Dounia
- ANAAM Said
- ANBRI Khalid
- BENMHAMED Chakib
- BOUTGOULA Hasna
- BOUYA OUSSAMA
- CHAOU Fatima Ezzahra
- ESSAKINE Ibrahim
- HARA Ikram
- LAMHAMDI ANOUAR
- MIQRI Hajar
- MOUNIR Mohamed
- OUHAGUA Omar
- ZAKI Sara



Encadré Par :

- Dr EL ALAOUI Mohamed

## Remerciements

Nous remercions tout d'abord Dieu tout puissant de nous avoir permis de mener à terme ce projet qui présente pour nous le point de départ de notre carrière.

Nous tenons aussi exprimer nos plus sincères remerciements au Professeur Mohamed El Alaoui, pour le soutien et l'encadrement tout au long de notre projet de VRP dans le cadre du module de logistique et transport.

Cher professeur vos conseils avisés et votre expertise nous ont permis de mieux comprendre les enjeux du sujet et de réaliser un travail de qualité.

Nous avons grandement apprécié votre disponibilité et votre engagement à nous aider à réussir ce projet. Votre implication a été un véritable atout pour notre équipe et nous sommes fiers de vous avoir comme professeur.

Encore une fois, merci beaucoup pour tout ce que vous avez fait pour nous. Un grand merci à l'équipe du projet et les amis pour leur présence et soutien inconditionnels, et tous ceux et celles qui au long de ce travail qui nous ont soutenu.

Finalement, nous espérons que notre travail vous procure une parfaite satisfaction et soit à la hauteur

Cordialement,

# Table des matières

|  |                                    |
|--|------------------------------------|
| <b>Remerciements .....</b>   | 1                                  |
| <b>Liste des figures .....</b>   | 4                                  |
| <b>Liste des tableaux .....</b>  | 5                                  |
| <b>Introduction générale : .....</b>   | 6                                  |
| <b>Scénario.....</b>   | <b>Erreur ! Signet non défini.</b> |
| <b>Partie 1 : Le problème de la tournée de véhicule .....</b>                              | 8                                  |
| I.    Introduction.....  | 9                                  |
| II.   La Logistique et le PTV .....  | 10                                 |
| 1)   Types de problèmes de la tournée de Véhicule (VRP).....                               | 11                                 |
| 2)   Problème de la tournée de véhicule avec contrainte de capacité (exemple choisi) ..... | 12                                 |
| III.  Conclusion .....   | 17                                 |
| <b>Partie 2 : Les Algorithmes de résolution du VRP .....</b>                               | 18                                 |
| 1)   Introduction.....   | 19                                 |
| 2)   L'algorithme Branch and Bound.....  | 19                                 |
| 1)   Etapes de l'algorithme : .....  | 20                                 |
| 2)   Exemple d'application : .....   | 21                                 |
| 3)   L'algorithme du plus proche voisin : .....  | 23                                 |
| 1.   Les étapes de l'algorithme du plus proche voisin.....                                 | 23                                 |
| 2.   Discussion et résultat : .....  | 23                                 |
| 3.   Traitement du PRV avant d'utiliser l'algorithme du plus proche voisin .....           | 25                                 |
| 4.   L'étude du PRV en utilisant l'algorithme du plus proche voisin.....                   | 26                                 |
| 5.   Résumé : .....  | 26                                 |
| 4)   L'algorithme génétique :.....   | 27                                 |
| 1.   Définition :.....   | 27                                 |
| 2.   Justification du choix : .....  | 27                                 |

|      |  |           |
|------|--|-----------|
| 3.   | Historique : .....   | 27        |
| 4.   | Algorithme : Pseudo-Code d'un algorithme génétique de base.....            | 28        |
| 5.   | Exemple d'explication :.....   | 31        |
| 6.   | Conclusion :.....  | 32        |
|      | <b>Partie 3 : Les solutions informatique de résolution du VRP .....</b>    | <b>33</b> |
| I.   | Introduction.....  | 34        |
| II.  | Solution exacte proposés Pulp :.....                                       | 34        |
| 1.   | Bibliothèques utilisées.....   | 34        |
| 2.   | Implantation :.....  | 36        |
| III. | Solution informatique en utilisant l'outil : OpenDoorLogistics Studio..... | 45        |
| 1.   | Présentation de OpendoorLogistic Studio : .....                            | 45        |
| 2.   | Raisons du choix de OpenDoorLogistics Studio : .....                       | 45        |
| 3.   | Résolution :.....  | 46        |
| 4.   | Extensions : .....   | 53        |
| IV.  | Outil exakte --Open Solver (COIN-OR CBC)-- .....                           | 54        |
| 1.   | Énoncé du problème : .....   | 54        |
| 2.   | Données :.....   | 54        |
| 3.   | Méthodologie : .....   | 54        |
| 4.   | Résolution :.....  | 59        |
| 5.   | Conclusion:.....   | 65        |
|      | <b>Conclusion générale :</b> .....   | <b>66</b> |

# Liste des figures

|  |                                    |
|--|------------------------------------|
| FIGURE 1 MATRICE DES DISTANCES REGION 1 .....                                    | 36                                 |
| FIGURE 2 REPRESENTATION DES CLIENTS ET DEPOT REGION 1 .....                      | 36                                 |
| FIGURE 3 CORDONNEES DES PROVINCES DE REGION 1 .....                              | 37                                 |
| FIGURE 4 REPRESENTATION DE LA SOLUTION DE REGION 1 .....                         | 37                                 |
| FIGURE 5 MATRICE DES DISTANCES DE REGION 2 .....                                 | 38                                 |
| FIGURE 6 REPRESENTATION DES CLIENTS ET DEPOT REGION 2 .....                      | 38                                 |
| FIGURE 7 REPRESENTATION DE LA SOLUTION DE REGION 2.....                          | 39                                 |
| FIGURE 8 MATRICE DES DISTANCES DE REGION 3 .....                                 | 39                                 |
| FIGURE 9 REPRESENTATION DES CLIENTS ET DEPOT REGION 3 .....                      | 40                                 |
| FIGURE 10 REPRESENTATION DE LA SOLUTION DE REGION 3. ....                        | 41                                 |
| FIGURE 11 MATRICE DES DISTANCES DE REGION 3 .....                                | 41                                 |
| FIGURE 12 REPRESENTATION DES CLIENTS ET DEPOT REGION 3.....                      | 42                                 |
| FIGURE 13 REPRESENTATION DE LA SOLUTION DE REGION 3.....                         | 42                                 |
| FIGURE 14 MATRICE DES DISTANCES DE REGION 3 .....                                | 43                                 |
| FIGURE 15 REPRESENTATION DES CLIENTS ET DEPOT REGION 3 .....                     | 44                                 |
| FIGURE 16 REPRESENTATION DE LA SOLUTION DE REGION 3. ....                        | 44                                 |
| FIGURE 17 OPENDOOR LOGISTIC STUDIO.....  | <b>ERREUR ! SIGNET NON DEFINI.</b> |
| <b>FIGURE 18 : PAGE DE TELECHARGEMENT D 'OPENDOORLOGISTICS STUDIO .....</b>      | 46                                 |
| FIGURE 19 CREATION DE NOUVEAU FICHIER.....                                       | 47                                 |
| FIGURE 20 :MENU D'INSTRUCTIONS UTILES .....                                      | 49                                 |
| FIGURE 21 CREATION DES TABLEAUX DE DONNEES .....                                 | 50                                 |
| FIGURE 22 :INPUTS TABLES .....   | 50                                 |
| FIGURE 23 GENERATION DES GEOCODE DE CHAQUE PROVINCE UTILISANT GOOGLE SHEET ..... | 51                                 |
| FIGURE 24 FICHIER INPUT POUR LES PROVINCES DU MAROC .....                        | 51                                 |
| FIGURE 25 CHARGEMENT D'INPUT TABLE SUR ODL STUDIO .....                          | 52                                 |
| FIGURE 26 OPTIMISER.....   | 52                                 |
| FIGURE 27 PARAMETRAGE DE OPEN SOLVER.....  | 55                                 |
| FIGURE 28 MATRICE DES CONTRAINTES .....  | 56                                 |
| FIGURE 29 MATRICE DES DISTANCES .....  | 56                                 |
| FIGURE 30 MATRIX BOOLEAN .....   | 56                                 |
| FIGURE 31 MATRIX FLOW .....  | 57                                 |
| FIGURE 32 : ENTRER LE MODELE DANS OPEN SOLVER. ....                              | 57                                 |
| FIGURE 33 REPRESENTATION GRAPHIQUE DE LA SOLUTION .....                          | 59                                 |

## Liste des tableaux

|           |  |    |
|-----------|--|----|
| TABLEAU 1 | LES DONNEES DES CLIENTS A SERVIR ..... | 24 |
| TABLEAU 2 | . DISTANCE ET HEURE DE LIVRAISON.....  | 24 |
| TABLEAU 3 | LES PARAMETRES DES VEHICULES.....      | 25 |

## Introduction générale :

L'évolution du marché mondial, caractérisée par la diversification des sources d'approvisionnement, la spécialisation des sites de production, les échanges inter-usines et parfois même internationaux, la création de sites de stockage intermédiaires et la commercialisation des produits, fait apparaître de nouveaux problèmes auxquels le manager doit faire face. En outre, les débats politiques sur la protection de l'environnement et la pollution par les émissions des gaz à effet de serre (CO, CO<sub>2</sub> . . .) et le pétrole qui se fait de plus en plus rare ainsi que l'augmentation des prix de revient des énergies non renouvelables ne font que renforcer l'inquiétude des managers. Plus de 30% des dépenses d'une chaîne logistique sont consacrés au transport, ce taux ne fera qu'augmenter avec le temps.

Donc l'optimisation de la chaîne de transport est plus que vitale pour l'épanouissement de toute économie. C'est le paramètre de base, le secret de la réussite, sur lequel le manager doit investir pour se distinguer des autres. Il doit innover sans cesse, créer, des méthodes permettant l'organisation optimale de sa chaîne de transport en concordance avec l'ensemble de sa chaîne logistique.

Le problème de transport est célèbre en optimisation combinatoire. Le problème de base et probablement le plus étudié, est le Problème du Voyageur de Commerce (PVC) dit aussi Traveling Salesman Problem (TSP), qui a pour objectif, la visite aux moindres coûts d'un ensemble de clients, une et une seule fois, avec un seul véhicule.

L'extension du nombre de véhicules d'un à une flotte de véhicules définit un autre problème de transport. Il est dit Problème de Tournées de Véhiculées ou Vehicul Routing Problem (VRP). L'enrichissement de ce problème se fait par des contraintes relatives au nombre de véhicules, à leurs charges, ou par des contraintes relatives aux clients, à leurs demandes, leurs fenêtres temporelles, définit de nombreuses variantes.

L'objectif de ce travail est la résolution d'un problème de tournées de véhicules classique en utilisant un ensemble de méthodes qui s'appuient sur des algorithmes exactes et approchées,

heuristiques et Meta heuristiques et en présentant quelques outils informatiques visant à optimiser le trajet, le temps et la distance de la tournée.

Dans la première partie, nous présentons les concepts de base de l'optimisation combinatoire, en particulier le VRP. Ensuite, nous présentons un état d'art du VRP et sa formulation mathématique. La deuxième partie est consacrée aux méthodes de résolution de problème en utilisant l'algorithme de Branch and Bound, le plus proche voisin et l'algorithme génétique. La troisième partie fait l'objet d'une description de l'environnement de programmation et l'implémentation de l'ensemble des solutions informatique proposées et réalisées. Le travail est clôturé par une analyse des résultats et une conclusion dans laquelle nous évaluons l'approche de résolution programmée et donnons quelques des perspectives de recherche.

## **Partie 1 : Le problème de la tournée de véhicule**

## I. Introduction

Le transport est indispensable à notre économie en rendant possible les déplacements de produits entre les différents maillons de la chaîne logistique. Son rôle prend tout son poids dans plusieurs pays. Et les entreprises ayant à transporter beaucoup de biens avec une bonne planification des transports comme un grand facteur de rentabilité. Donc le fait d'assigner et d'ordonnancer les livraisons à des tournées de véhicules permet d'impacter considérablement les coûts d'une entreprise. C'est la raison pour laquelle il y a un grand intérêt soutenu de la part des chercheurs pour les problèmes de transport, et plus particulièrement le problème de tournée de véhicules (PTV).

Le PTV ou VRP (Vehicle Routing Problem) dans sa version classique consiste à déterminer comment un ensemble de clients peut être desservi par une flotte de véhicules homogène basée à un dépôt. Tout en minimisant les coûts de transport qui peuvent être définis en temps, en coût monétaire ou en distance parcourue.

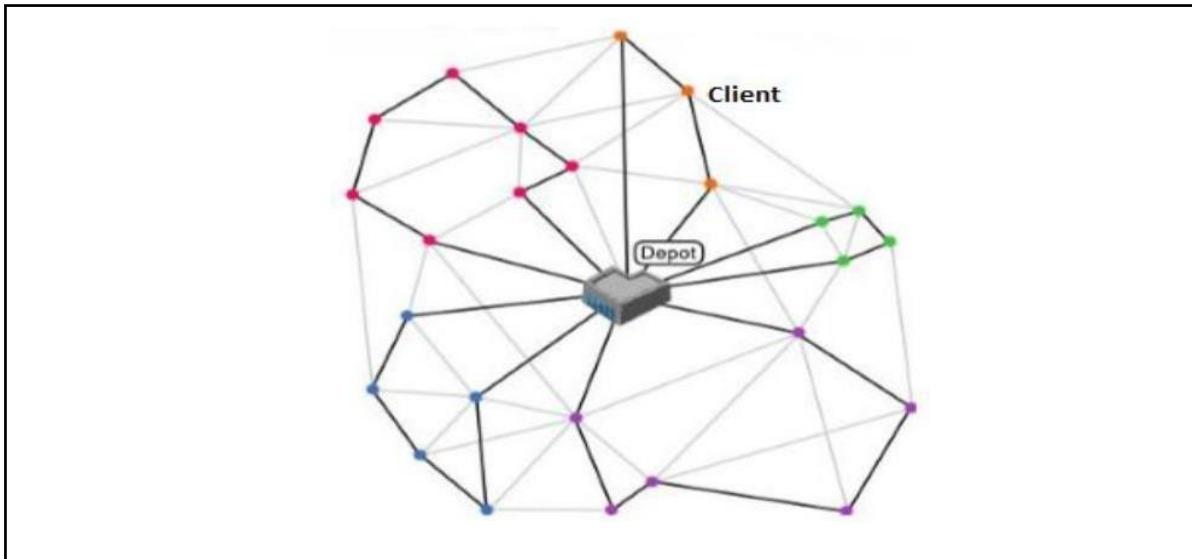
En recherche opérationnelle, le PTV a aussi été étudié sous des formes spécifiques permettant de modéliser des problèmes plus près de la réalité. Car il y a un ensemble de paramètres qui ont été modifiés ou ajoutés au problème classique comme, la possibilité d'avoir une flotte de véhicules hétérogène, la présence de plusieurs dépôts ou bien encore des fenêtres de temps de livraison des clients.

## II. La Logistique et le PTV

La logistique est l'ensemble des activités qui permettent d'optimiser la gestion des flux de marchandises, d'informations et de personnes, de la production à la livraison finale en passant par la distribution. Elle regroupe ainsi les différentes étapes de la chaîne d'approvisionnement, de l'achat des matières premières jusqu'à la livraison des produits finis.

Le VRP (Vehicle Routing Problem) quant à lui est un problème mathématique relevant de l'optimisation combinatoire, qui consiste à déterminer les meilleurs trajets pour un ensemble de véhicules afin de livrer une série de points de livraison en minimisant la distance totale parcourue. Le VRP est donc une sous-discipline de la logistique qui s'intéresse plus spécifiquement à la gestion des tournées de véhicules.

Ainsi, le VRP est un outil de la logistique qui permet d'optimiser la gestion des flux de marchandises en déterminant les meilleurs trajets pour les véhicules de livraison. Il s'agit d'un enjeu important pour les entreprises, car la gestion optimale des tournées permet de réduire les coûts de transport, d'améliorer la qualité de service en respectant les délais de livraison et de réduire l'impact



environnemental en réduisant les distances parcourues.

Fig 1 : Présentation graphique du problème de tournées de véhicules VRP

## 1) Types de problèmes de la tournée de Véhicule (VRP)

Durant des années de recherche sur le VRP d'autres dérivées de ce problème sont apparues. Ces apparitions sont dues essentiellement aux activités des chercheurs qui travaillent de plus sur les problèmes de transport et de distribution que rencontrent les sociétés. Dans ce qui suit nous allons présenter les principaux problèmes dérivés du PTV :

### a. PTVC (*Problème de tournées de véhicules avec capacité*)

Un problème PTVC consiste à affecter chaque client à une tournée effectuée par un seul véhicule de capacité finie. Ce véhicule commence et termine sa tournée au dépôt.

### b. PTVD (*Problème de tournées de véhicule dynamique*)

Contrairement au VRP classique où toutes les demandes des clients sont connues et planifiées à l'avance, le PTVD illustre mieux la réalité. Dans les problèmes de transports réels dans la société d'aujourd'hui la composante dynamique est très importante, ou de pannes des véhicules de transport. Dans ce genre de situation, le décideur doit réorganiser, en un temps acceptable, la planification des itinéraires de sa flotte de véhicules pour répondre aux nouvelles demandes urgentes.

### c. VRPB (*Prolongation du VRP classique*)

Ce problème est une prolongation du VRP classique. Il comporte des clients auxquels des produits doivent être livrés et des fournisseurs dont les marchandises doivent être amenées au dépôt. Donc, pour éviter de réarranger les produits transportés dans le véhicule sur une tournée, les clients doivent être visités avant les fournisseurs.

### d. PTVMP (*Problème de tournées de véhicules multi périodique*)

Le problème de tournées de véhicules multi périodique (PTVMP) consiste à livrer pour un ensemble de clients, la quantité demandée d'un ou de plusieurs produits sur un horizon de temps donné. Dans ce problème, la quantité de produits livrée à un client, permet à ce dernier de subvenir à ses besoins en attendant la prochaine visite du véhicule. Le but principal de ce problème se divise en deux parties : la première consiste à planifier les horaires de livraison de chaque client sur un horizon de temps prédéterminé ; la deuxième consiste à organiser les tournées des véhicules afin d'effectuer les livraisons nécessaires tout en optimisant le coût total de transport. Outre la

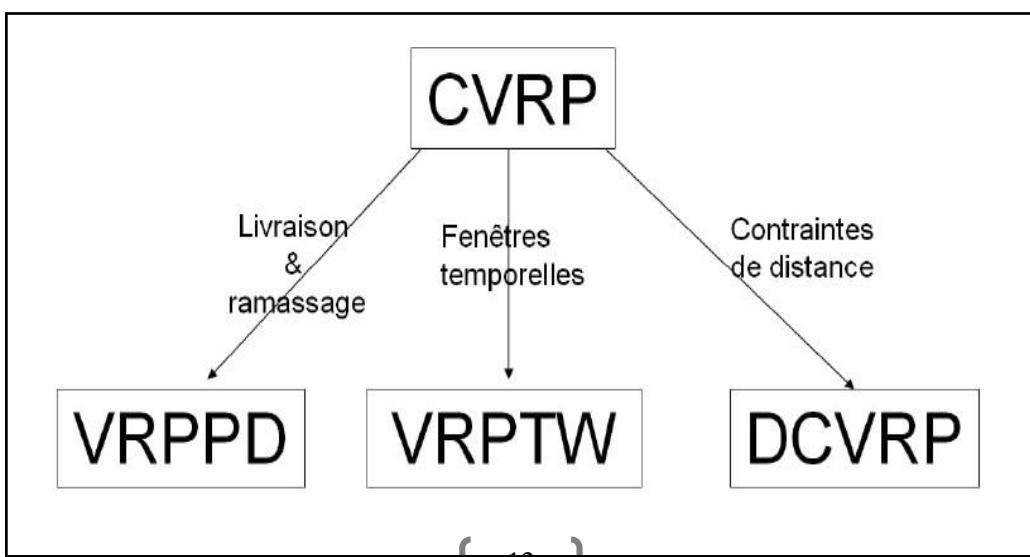
connaissance et l'utilisations des contraintes connues d'un problème de transport, le PTVMP nécessite l'intégration des éléments de gestion de stock afin de respecter les besoins en quantité et en temps des clients.

e. *PTVS (Problème de tournée des véhicules stochastique)*

Un problème VRP est dit stochastique si au moins un de ses éléments est aléatoire, c'est à dire avec une certaine incertitude. Ces éléments peuvent être une ou plusieurs demandes des clients, les temps ou les couts de transport. Le problème avec les demandes stochastique est celui le plus étudie dans la littérature. La majorité des recherches suppose que les demandes aléatoires suivent une loi de distribution normale.

## 2) Problème de la tournée de véhicule avec contrainte de capacité (exemple choisi)

Le Problème de Tournées de Véhicules avec Capacités (CVRP) est un problème d'optimisation de tournées de véhicules avec des contraintes sur les capacités des véhicules. Dans un programme CVRP, chaque emplacement est associé à une demande de marchandises à ramasser ou à livrer. Chaque fois qu'un véhicule visite un emplacement, la quantité totale de marchandises qu'il transporte augmente (en cas de ramassage) ou diminue (en cas de livraison) en fonction de la demande de l'emplacement visité. Un graphe peut être utilisé pour représenter un CVRP, dans lequel les arêtes représentent les distances et les nœuds correspondent aux demandes. Si chaque client a une demande fixe et chaque véhicule a une capacité de stockage limitée, alors il s'agit d'un CVRP (Capacité VRP). Le problème est alors compliqué par la contrainte de limiter la demande cumulée des emplacements visités à la capacité de stockage du véhicule parcourant le circuit.



### *a. Expression de besion (Scénario)*

Notre entreprise de transport s'est vu confier une mission spéciale : livrer des produits dans toutes les villes du Maroc en un temps record. Pour accomplir cette tâche, nous avons utilisé un modèle innovant de résolution de problème de tournée de véhicules (CVRP) qui a permis d'optimiser notre trajet et d'économiser du temps et de l'argent.

Notre algorithme CVRP a d'abord pris en compte toutes les villes du Maroc que nous devions visiter et les distances entre chacune d'elles. Nous avons ensuite optimisé notre itinéraire en utilisant des algorithmes de recherche locale pour minimiser la distance parcourue et réduire le nombre de véhicules nécessaires. Nous avons également pris en compte les restrictions de temps pour assurer une livraison rapide et fiable.

Notre entreprise a ensuite mis en œuvre cette solution innovante en utilisant des camions électriques écologiques pour réduire notre empreinte carbone. En outre, nous avons travaillé en étroite collaboration avec les autorités locales pour assurer la fluidité de la circulation dans les villes et minimiser les retards.

Grâce à notre modèle CVRP, nous avons réussi à livrer toutes les villes du Maroc en un temps record et à réduire considérablement nos coûts de transport. Cette solution innovante peut être adaptée à toute entreprise de transport souhaitant améliorer son efficacité et sa durabilité environnementale.

### *b. Objectif*

L'objectif du Problème de Routage de Véhicules (PRV) et du Problème de Routage de Véhicules avec Capacités (PRVC) est de minimiser le coût total du voyage, c'est-à-dire la somme des distances ou des temps de parcours des tournées. Toutefois, pour le PRVC, la capacité de chaque véhicule doit être prise en compte si la somme de toutes les demandes est dynamique et si de nouvelles exigences se présentent. Le VRP est l'un des problèmes les plus difficiles qui a suscité beaucoup d'attention et de nombreuses études. Il a de nombreuses applications dans la vie réelle et trouver une solution optimale est un effort continu. En d'autres termes, l'objectif est de minimiser le coût total tout en respectant la contrainte de capacité des véhicules : la quantité de marchandises livrées sur une tournée ne doit pas dépasser la capacité du véhicule qui assure la tournée.

Le problème de routage est décrit comme suit :

- ✓ Chaque véhicule sert un ensemble de clients et revient à :
- ✓ Un certain nombre de véhicules proviennent d'un seul dépôt.
- ✓ L'objectif est de réduire la distance totale parcourue.
- ✓ La demande totale des clients desservie par un véhicule doit être le même dépôt
- ✓ Tous les clients sont desservis par un seul véhicule.
- ✓ Ne pas dépasser la capacité du véhicule.

### c. La formulation mathématiques VRP

Formulation La version standard du VRP considère une contrainte de capacité (PTVC, Problème De Tournée Des Véhicules Avec Capacité) Il peut être représenté sous la forme d'un graphe orienté et value  $G = (N, A)$

Où :

- ✓  $N$  représente les positions des clients et du dépôt,
- ✓  $A$  représente les arcs entre deux clients  $i, j \in N$ .

Plus spécifiquement, nous avons un ensemble  $C = \{1, \dots, nc\}$  de clients qui doivent obtenir une livraison de merchandise provenant du dépôt. L'ensemble des positions de ces clients ou nœuds est défini par l'ensemble  $N = C \cup \{0, nc+1\}$  où 0 et  $nc+1$  représente le dépôt (aller et retour).

Une demande positive de produit  $d_i$  est associée à chaque client  $i$  appartenant à  $C$ .

Une flotte de véhicules  $V = \{1, \dots, nv\}$  est disponible au dépôt et chaque véhicule possède la même capacité (flotte homogène)  $Q$  telle que  $Q \geq \max_i d_i, \forall i \in N$ . Pour tous les clients  $i$  et  $j$ ,  $\forall i, j \in N$ , nous connaissons le cout  $c_{ij}$  de transport direct entre  $i$  et  $j$  (proportionnel à la distance à parcourir). Pour trouver l'ordre de visite des clients, nous définissons les variables de décisions comme suit :

$x_{v,i} = \begin{cases} 1 & \text{Si le Véhicule } v \in V \text{ visite le client } i \text{ après le client } j, \\ 0 & \text{Sinon} \end{cases}$

En définissant  $y_i$  comme étant la charge résiduelle du véhicule après avoir desservi le client  $i \in C$ . Il nous est possible d'écrire formellement le modèle de VRP. Il s'agit d'optimiser la fonction suivante :

$$\text{Min} \sum_{v \in V} \sum_{i \in N} \sum_{j \in N} C_{i,j} x_{i,j}^v \quad (1)$$

$$v \in V \quad i \in N \quad j \in N$$

Avec les contraintes :

$$\sum_{v \in V} \sum_{i \in N} x_{i,j}^v = 1, \forall i \in C \quad (2)$$

$$v \in V \quad j \in N$$

$$\sum_{i \in N} x_{i,h}^v - \sum_{j \in N} x_{h,j}^v = 0, \forall h \in C, v \in V \quad (3)$$

$$\sum_{j \in N} x_{0,j}^v = 1, \forall v \in V \quad (4)$$

$$j \in N$$

$$\sum_{j \in N} x_{j,n+1}^v = 1, \forall v \in V \quad (5)$$

$$x_{ij}^v = 1 \rightarrow y_i - d_j = y_j, \forall i, j \in N, v \in V \quad (6)$$

$$y_0 = Q, 0 \leq y_i, \forall i \in C \quad (7)$$

$$x_{ij}^v \in \{0,1\}, \forall i, j \in N, v \in V \quad (8)$$

La fonction de cout de la solution  $X = (xv), \forall i, j \in N, \forall v \in V$  est définie par :

$$\text{cout}(X) = \sum_{i,j} \sum_{C_{i,j}x^v} \quad (9)$$

Le nombre de véhicule utilisés par la solution X, est défini par :

$$Nb \text{ véhicule } (X) = \sum \sum x_{0,j} \quad (10)$$

La fonction objective (équation 1) représente le nombre de véhicules utilisés pour les trajets effectués et la somme des couts s'y rapportant.

La formulation du problème nécessite de faire certaines contraintes :

- ✓ L'équation 2 assure qu'on part une et une seule fois de chaque client, avec un seul véhicule.
- ✓ L'équation 3 assure que le véhicule qui arrive chez un client est le même que celui qui part de ce client.
- ✓ L'équation 4 assure que chaque véhicule ne sort qu'une seule fois du dépôt.
- ✓ L'équation 5 assure le retour unique au dépôt pour chaque véhicule (ou tournée). Il n'y a pas de restriction sur le nombre de véhicule, mais un cout  $c_v$  est affecté à chaque véhicule utilisé. On impose une valeur de  $c_v$  suffisamment grande pour réduire principalement le nombre de véhicule et pour réduire au minimum dans un deuxième temps les couts de transport.
- ✓ Les équations 6 et 8 définissent les contraintes de capacité et d'intégrité.
- ✓ Les équations 9 et 10 sont des fonctions de mesure qui permettent respectivement de quantifier la solution selon la distance totale parcourue. Ainsi que le nombre de véhicules utilisés.

Le problème d'élaboration de tournées de véhicule est un problème NP-difficile, c'est-à-dire qu'il n'existe pas à ce jour un algorithme déterministe pouvant résoudre ce problème en temps polynomial.

Pour des problèmes comportant un grand nombre de clients (<100 clients), des méthodes approchées sont nécessaires pour les résoudre.

### III. Conclusion

Nous avons présenté dans ce chapitre les principaux concepts de problème de la tournée des véhicules, nous avons vu que cette problématique est dévisée en plusieurs types. Le chapitre suivant on présentera les méthodes de résolution du problème de VRP, et on va entamer quelques algorithmes à savoir (exacte : Branch and Bound, heuristique : le plus proche voisin, méta-heuristique : méthode génétique).

## **Partie 2 : Les Algorithmes de résolution du VRP**

## 1) Introduction

Dans le contexte du routage des véhicules, plusieurs algorithmes sont utilisés pour résoudre les problèmes de manière efficace. Trois approches couramment utilisées sont l'algorithme Branch and Bound, l'algorithme du Plus Proche Voisin et l'algorithme Génétique. Dans ce deuxième chapitre, nous allons examiner en détail ces trois algorithmes et leur application dans la résolution des problèmes de routage des véhicules.

L'algorithme Branch and Bound est une méthode de recherche exacte qui explore systématiquement l'espace des solutions en utilisant une stratégie de découpage et de relaxation. Il permet de trouver la meilleure solution possible en examinant toutes les solutions potentielles tout en évitant de revisiter les sous-espaces déjà explorés. Cela en fait une approche efficace pour les problèmes de petite à moyenne taille où l'optimalité de la solution est primordiale.

L'algorithme du Plus Proche Voisin est une méthode heuristique simple qui commence à partir d'un point de départ et sélectionne itérativement le point suivant le plus proche non encore visité jusqu'à ce que tous les points soient visités. C'est une approche rapide et facile à implémenter qui donne souvent des résultats acceptables en termes de qualité de solution pour les problèmes de routage des véhicules.

L'algorithme Génétique est une méthode de recherche métaheuristique inspirée des principes de l'évolution biologique. Il utilise des opérations génétiques telles que la mutation, la recombinaison et la sélection pour générer une population de solutions et les améliorer progressivement au fil des générations. C'est une approche efficace pour les problèmes de grande taille et complexes, mais qui nécessite souvent plus de temps de calcul.

## 2) L'algorithme Branch and Bound

L'algorithme de Branch and Bound est un algorithme utilisé pour résoudre des problèmes de recherche et d'optimisation. Il s'agit d'une méthode qui explore l'arbre des possibilités. L'algorithme utilise un parcours en largeur pour explorer l'arbre, et cherche à estimer un intervalle aussi étroit

que possible pour la longueur d'un plus court chemin obtenu à partir de la branche de l'arbre où nous sommes.

L'algorithme compare ensuite les intervalles obtenus pour élaguer les parties de l'arbre qui ne sont pas utiles. Par exemple, si une branche a une borne supérieure plus petite qu'une autre branche ayant une borne inférieure plus grande, alors l'algorithme élimine le sous-arbre correspondant à la deuxième branche.

En somme, l'algorithme de Branch and Bound est une méthode efficace pour résoudre des problèmes de recherche et d'optimisation en explorant l'arbre des possibilités de manière optimale, grâce à l'utilisation de bornes pour élaguer les branches inutiles.

### 1) Etapes de l'algorithme :

#### a) *Etape de relaxation:*

Dans cette étape, le problème d'optimisation initial est relaxé pour permettre la création d'une borne supérieure et inférieure pour la solution optimale. Cela permet de déterminer rapidement si une branche donnée peut fournir une solution meilleure que la borne actuelle. Pour le VRP, cela pourrait impliquer de supposer que tous les clients peuvent être desservis par un seul véhicule.

#### b) *Étape de sélection de branche :*

Cette étape consiste à choisir une branche à explorer à partir de l'arbre de recherche. La branche sélectionnée est généralement celle qui a la plus grande chance d'offrir une meilleure solution. Pour le VRP, cela pourrait impliquer de choisir un nœud dans l'arbre qui représente l'affectation d'un véhicule à un client spécifique.

#### c) *Étape de résolution :*

Dans cette étape, le sous-problème associé à la branche sélectionnée est résolu en utilisant une méthode exacte ou heuristique. Cette étape permet de générer une nouvelle solution candidate. Pour le VRP, cela pourrait impliquer de résoudre le problème d'affectation pour un sous-ensemble de clients et de véhicules.

*d) Étape de mise à jour des bornes :*

Les bornes supérieure et inférieure sont mises à jour en fonction de la nouvelle solution générée. Si la nouvelle solution est meilleure que la borne supérieure actuelle, la borne supérieure est mise à jour. Si la nouvelle solution est pire que la borne inférieure actuelle, la branche est abandonnée. Pour le VRP, cela pourrait impliquer de mettre à jour la borne supérieure en fonction de la meilleure solution trouvée jusqu'à présent pour l'ensemble des clients et des véhicules.

*e) Étape de terminaison :*

Cette étape consiste à vérifier si la condition d'arrêt a été satisfaite. Si la condition est satisfaite, l'algorithme s'arrête et la solution optimale est renvoyée. Sinon, l'algorithme continue à explorer les branches non explorées, nous répétons les étapes 2 à 4 jusqu'à ce que la solution optimale soit trouvée. Pour le VRP, cela pourrait impliquer de vérifier si toutes les affectations de véhicules à des clients ont été explorées et si la meilleure solution trouvée jusqu'à présent répond aux critères d'arrêt, tels que le temps de calcul ou l'écart de la solution optimale.

## 2) Exemple d'application :

Prenons l'exemple d'un VRP avec les données suivantes :

- ✓ Un dépôt situé en (0,0)
- ✓ 4 clients à visiter situés aux positions (1,2), (2,5), (5,2) et (6,6)
- ✓ Capacité de la voiture = 7

Nous pouvons résoudre ce problème en utilisant l'algorithme de Branch and Bound de la manière suivante :

### Initialisation :

- ✓ UB = infini
- ✓ LB = 0
- ✓ Créer le nœud racine avec la solution vide
- ✓ Ajouter le nœud racine à la liste des nœuds actifs

### Sélection du nœud à développer :

- ✓ Utiliser la stratégie BFS pour sélectionner le nœud le plus prometteur
- ✓ Supposons que le nœud sélectionné soit le nœud [0]

### **Division du nœud sélectionné :**

- ✓ Diviser le nœud [0] en deux sous-nœuds : le sous-nœud [0,1] avec le client 1 attribué à la même voiture et le sous-nœud [0,2] avec le client 1 attribué à une nouvelle voiture.

### **Évaluation des sous-nœuds :**

- ✓ Évaluer le sous-nœud [0,1] : le coût est de 5 (distance entre le dépôt et le client 1) + 2 (distance entre le client 1 et le dépôt) = 7. Le sous-nœud respecte la contrainte de capacité.
- ✓ Évaluer le sous-nœud [0,2] : le coût est de 5 (distance entre le dépôt et le client 1) + 2 (distance entre le client 1 et le client 2) + 3 (distance entre le client 2 et le dépôt) = 10. Le sous-nœud respecte la contrainte de capacité.

### **Mise à jour des bornes :**

- ✓ LB est mis à jour à la valeur minimale des coûts des sous-nœuds évalués, soit 7.
- ✓ UB reste à infini, car nous n'avons pas encore trouvé de solution complète.

### **Ajout des nœuds actifs :**

- ✓ Ajouter les sous-nœuds [0,1] et [0,2] à la liste des nœuds actifs, triée par ordre croissant de coût.

### **Répéter les étapes 2 à 6 jusqu'à ce que la liste de nœuds actifs soit vide.**

- ✓ Nous développerons le sous-nœud [0,1], puis les sous-nœuds [0,2,3] et [0,2,4] pour continuer la recherche.
- ✓ Le sous-nœud [0,1] ne peut plus être développé car tous les clients ont été visités. Nous mettons à jour UB avec la valeur 7, car c'est la meilleure solution trouvée jusqu'à présent.
- ✓ Le sous-nœud [0,2,3] ne respecte pas la contrainte de capacité, il est élagué.
- ✓ Le sous-nœud [0,2,4] peut être développé. Nous obtenons la solution complète avec un coût de 7 (distance entre le dépôt et le client 1) + 4 (distance entre le client 1 et le client 3) + 3 (distance entre le client 3 et le dépôt) = 14.

### **Terminaison :**

- La liste de nœuds actifs est vide. La recherche est terminée.
- La meilleure solution trouvée a un coût de 14.

### **La solution trouvée est donc la suivante :**

- ✓ La voiture 1 visite les clients 1 et 3 dans cet ordre, avec un coût de 7.
- ✓ La voiture 2 visite les clients 2 et 4 dans cet ordre, avec un coût de 7.
- ✓ Le coût total est de 14.

**Remarque :** Cet exemple est simplifié et ne prend pas en compte toutes les combinaisons possibles. En pratique, l'algorithme de Branch and Bound peut explorer un grand nombre de combinaisons et il est donc important d'optimiser les stratégies de sélection des nœuds à développer pour réduire le temps de calcul.

### 3) L'algorithme du plus proche voisin :

Dans cette partie on va utiliser l'algorithme du plus proche voisin comme méthode heuristique pour résoudre le VRP. C'est une méthode heuristique utilisée pour résoudre des problèmes d'optimisation combinatoire tels que le VRP. L'objectif est de trouver une solution réalisable en un temps raisonnable, plutôt que d'essayer de trouver la solution optimale en un temps très long.

L'algorithme commence par sélectionner un point de départ et recherche ensuite le point le plus proche qui n'a pas encore été visité. Ce point devient alors la prochaine destination. Le processus se répète jusqu'à ce que tous les clients soient desservis.

Bien que l'algorithme du plus proche voisin ne garantisse pas une solution optimale, il est simple et rapide à mettre en œuvre, ce qui le rend utile pour résoudre des problèmes de grande taille. Dans l'exemple qu'on va traiter dans cette partie, l'algorithme du plus proche voisin permet d'économiser la distance de 13.14% et le coût avec un taux de 13.17%.

#### 1. Les étapes de l'algorithme du plus proche voisin

- ✓ Sélectionnez un point qui représente la ville ou le lieu de départ.
- ✓ Ensuite, sélectionnez le point de destination ou la ville suivante, en ne considérant que la ville ayant la distance la plus proche par rapport à la ville précédemment visitée.
- ✓ Si toutes les villes ont été visitées ou que tous les points ont été reliés, fermez l'itinéraire du voyage en retournant à la ville d'origine (le point de départ).

#### 2. Discussion et résultat :

Les données utilisées dans cet exemple proviennent d'une étude littéraire obtenue à partir d'un essai intitulé "Implémentation de l'algorithme des fourmis pour l'optimisation du plus court

chemin" écrit par Ernawati en 2017. Dans cette étude on va utiliser les données suivantes pour résoudre le PTV avec l'algorithme du plus proche voisin :

*Le bureau JNE livre des marchandises à 6 points de différentes localisations avec 2 véhicules.*

| Numéro | Lieux de livraison                  | Code |
|--------|-------------------------------------|------|
| 1      | JNE office                          | X    |
| 2      | Instalasi Kusta dan Infeksi Endemik | A    |
| 3      | RSIA Siti Khadijah III              | B    |
| 4      | PT. Indolok Bakti                   | C    |
| 5      | Dipa Jaya Sejahtera                 | D    |
| 6      | Mal GTC Tanjung Bunga               | E    |
| 7      | Dinas Kelautan dan Perikanan        | F    |

Tableau 1                    Les données des clients à servir

*Les distances entre les points sont représentées dans le tableau ci-dessous :*

| Numéro | Location | Distance en Km | Période en minutes |
|--------|----------|----------------|--------------------|
| 1      | X-A      | 2.5 Km         | 9 m                |
| 2      | X-B      | 2.9 Km         | 8 m                |
| 3      | X-C      | 2.9 Km         | 8 m                |
| 4      | X-D      | 3.3 Km         | 9 m                |
| 5      | X-E      | 6.2 Km         | 16 m               |
| 6      | X-F      | 2.3 Km         | 8 m                |
| 7      | A-B      | 1.1 Km         | 4 m                |
| 8      | A-C      | 1.3 Km         | 4 m                |
| 9      | A-D      | 1.9 Km         | 6 m                |
| 10     | A-E      | 6.7 Km         | 16 m               |
| 11     | A-F      | 2.9 Km         | 9 m                |
| 12     | B-C      | 0.8 Km         | 2 m                |
| 13     | B-D      | 1.7 Km         | 6 m                |
| 14     | B-E      | 8.1 Km         | 19 m               |
| 15     | B-F      | 3 Km           | 8 m                |
| 16     | C-D      | 1.2 Km         | 4 m                |
| 17     | C-E      | 5.4 Km         | 13 m               |
| 18     | C-F      | 2.4 Km         | 7 m                |
| 19     | D-E      | 5.6 Km         | 11 m               |
| 20     | D-F      | 2.4 Km         | 7 m                |
| 21     | E-F      | 3.4 Km         | 8 m                |

Tableau 2 . Distance et heure de livraison

| Numéro | Flottes    | Marque       | Consommation de Pertamax par litre | Prix par litre   | Prix par kilomètre |
|--------|------------|--------------|------------------------------------|------------------|--------------------|
| 1      | Courrier 1 | Vario 150 cc | 52.9 km/liter                      | Rp 9.850,-/liter | Rp 186.2,-/km      |
| 2      | Courrier 2 | Vario 150 cc | 52.9 km/liter                      | Rp 9.850,-/liter | Rp 186.2,-/km      |

Tableau 3 Les paramètres des véhicules

Pertamax : est un type d'essence en Indonésie.

Vario 150 cc : est un modèle de moto de la marque "Honda"

Rp 9.850,-/liter : est une indication de prix en Indonésie, qui signifie "9.850 roupies indonésiennes par litre".

### 3. Traitement du PRV avant d'utiliser l'algorithme du plus proche voisin

Avant d'utiliser la méthode du plus proche voisin le problème a été résolu comme suit :

- ✓ Le trajet du premier courrier est le suivant : X → E → D → A → X
- ✓ Et dans ce cas la distance parcourue est : 6,2 km + 5,6 km + 1,9 km + 2,5 km = 16,2 Km.
- ✓ Le temps de trajet du courrier 1 est de 16 minutes + 11 minutes + 6 minutes + 9 minutes = 42 minutes.
- ✓ Le cout du carburant utilisé par le premier courrier est :  $16,2 \text{ km} \times \text{Rp } 186,2$ . donc de cout Rp 3.016,44.

Concernant le courrier 2 :

- ✓ Le trajet est le suivant : X → C → F → B → X
- ✓ Et dans ce cas la distance parcourue est : 2,9 km + 2,4 km + 3 km + 2,9 km = 11,2 Km.
- ✓ Le temps de trajet du courrier 2 est de 8 minutes + 7 minutes + 8 minutes + 8 minutes = 31 minutes.
- ✓ Le cout du carburant utilisé par le premier courrier est :  $11,2 \text{ km} \times \text{Rp } 186,2$ . donc de cout Rp 2.085,44.

Donc la distance totale parcourue par les deux courriers pour livrer les marchandises du bureau JNE est de  $16,2 \text{ km} + 11,2 \text{ km} = 27,4 \text{ km}$ . Et les frais pour le carburant pour les deux courriers sont de  $\text{Rp } 3\,016,44 + \text{Rp } 2\,085,44 = \text{Rp } 5\,101,88$ .

#### 4. L'étude du PRV en utilisant l'algorithme du plus proche voisin

- ✓ D'après le tableau 1, l'emplacement le plus proche du point X est A. Du point A, le courrier se rend au point B. Du point B, il se rend au point C et revient au point X.
- ✓ La distance parcourue est de 2,5 km + 1,1 km + 0,8 km + 2,9 km. Donc La distance totale est de 7,3 km.
- ✓ Le temps total pris par le courrier 1 du point X au point X est de 23 minutes. Le coût du carburant utilisé par le courrier 1 est de  $7,3 \text{ km} \times \text{Rp } 186,2 = \text{Rp } 1.357,8$ .
- ✓ La distance totale parcourue par les deux courriers pour livrer les marchandises du bureau JNE et revenir au bureau est de  $7,3 \text{ km} + 16,5 \text{ km} = 23,8 \text{ km}$ .
- ✓ Le temps de trajet total le plus long est de 42 minutes.
- ✓ Les frais encourus par le bureau JNE pour le carburant des deux courriers sont de  $\text{Rp. } 1.357,80 + \text{Rp. } 3.072,30 = \text{Rp. } 4.430,10$ .

#### 5. Résumé :

D'après les données obtenues avant et après l'utilisation du plus proche voisin, on peut déduire que :

- ✓ 1-L'algorithme du plus proche voisin permet de résoudre des problèmes tels que le PRV et les résultats sont plus optimaux.
- ✓ 2- Dans cet exemple, la distance parcourue a été réduite de 13,14% après l'utilisation de l'algorithme du plus proche voisin.
- ✓ 3- Le coût total a été réduit de 13,17%.

## 4) L'algorithme génétique :

### 1. Définition :

Les algorithmes génétiques appartiennent à la famille des algorithmes évolutionnistes. Leur but est d'obtenir une solution approchée à un problème d'optimisation, lorsqu'il n'existe pas de méthode exacte (ou que la solution est inconnue) pour le résoudre en un temps raisonnable. Les algorithmes génétiques utilisent la notion de sélection naturelle et l'appliquent à une population de solutions potentielles au problème donné.

### 2. Justification du choix :

Les algorithmes génétiques ont été imaginés pour résoudre des problèmes complexes, où il n'existe pas de méthode simple et logique pour trouver la solution.

Un algorithme génétique est capable de fournir de bonnes solutions aux problèmes pour lesquels on ne connaît pas de méthode de résolution ou dont la solution exacte est trop complexe pour être calculée en un temps raisonnable. On utilise ce genre d'outil pour planifier des tournées de véhicules (VRP), constituer des équipes de travail, implanter de manière optimale des points de vente dans une région, gérer des portefeuilles financiers et même pour trouver la meilleure forme à donner à la turbine d'un réacteur nucléaire, etc.

### 3. Historique :

Les principes fondamentaux de ces algorithmes ont été exposés par [Holland 1975]. Ces algorithmes s'inspirent du fonctionnement de l'évolution naturelle, notamment la sélection de Darwin, et de la reproduction selon les règles de Mendel. La sélection naturelle, que Darwin appelle “élément propulseur” de l'évolution, favorise les individus d'une population qui sont le mieux adaptés à un environnement. La sélection est suivie de la procréation. Elle est réalisée à l'aide de croisements et de mutations au niveau du patrimoine génétique des individus (ou “génotypes”) qui est constitué d'un ensemble de gènes. Ainsi deux individus “parents”, qui se croisent, transmettent une partie de leur patrimoine génétique à leurs descendants. Le génotype de l'enfant fait que celui-ci est plus ou moins bien adapté à l'environnement. S'il est bien adapté, il a une plus grande chance de se reproduire dans la génération future. Au fur et à mesure des générations, on sélectionne les

individus les mieux adaptés, et l'augmentation du nombre des individus bien adaptés fait évoluer la population entière vers un optimum (local ou global).

Dans les algorithmes génétiques, nous essayons de simuler le processus d'évolution d'une population. Nous partons d'une population de  $N$  solutions du problème représentées par des individus (chromosomes). Cette population choisie aléatoirement est appelée population parent. Le degré d'adaptation d'un individu à l'environnement est exprimé par la valeur de la fonction coût  $f(x)$  (ou fitness), où  $x$  est la solution que l'individu représente. Nous disons qu'un individu est d'autant mieux adapté à son environnement, que le coût de la solution qu'il représente est plus faible (pour un problème de minimisation). Au sein de cette population, intervient alors la sélection au hasard d'un ou deux parents qui produisent une nouvelle solution, à l'aide des opérateurs génétiques, tels que le croisement et la mutation. La nouvelle population, obtenue par le choix de  $N$  individus parmi les parents et les enfants de l'étape courante, est appelée génération suivante. En itérant ce processus, nous produisons une population plus riche en individus mieux adaptés. L'algorithme (1,1) montre le Pseudo-Code de l'algorithme génétique de base :

#### 4. **Algorithme :** Pseudo-Code d'un algorithme génétique de base

POP, POP' : deux tableaux de taille  $N$

Initialiser la population POP

Evaluer la population POP

POP' est VIDE

Rechercher  $x \mid f(x) = \min_i[f(x_i)]$ ,  $1 \leq i \leq N$

$f_{\min} = f(x)$

$x_{\min} = x$

#### **REPETER**

Evaluer la population POP

**REPETER** //phase de reproduction génétique

sélection

croisement

mutation

**JUSQU'A POP'** remplie par les nouveaux individus

Sélectionner la nouvelle population à partir de (POP, POP')

Rechercher  $x \mid f(x) = \min_i[f(x_i)]$ ,  $1 \leq i \leq N$

**SI**  $f(x) < f_{\min}$

$f_{\min} = f(x)$

$x_{\min} = x$

**FIN DE SI**

**JUSQU'A** conditions d'arrêt satisfaites

**Résultat (xmin, fmin)**

---

Où, POP représente la population courante et POP' représente la population des individus générés par l'étape de reproduction.

**Opérateurs de reproduction :** La phase de reproduction exploite principalement deux opérateurs : le croisement et la mutation. Elle comporte aussi l'opération de sélection et l'opération de production de la génération suivante, à partir des populations parent et enfant.

#### – Sélection

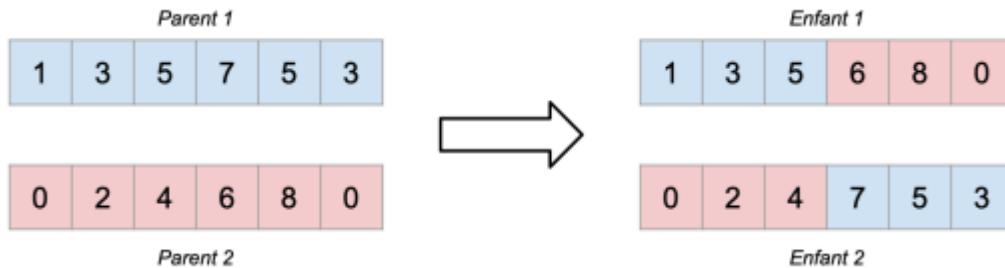
La sélection consiste à choisir les individus qui vont participer à la reproduction de la population future. La fonction de sélection choisit, de façon déterministe ou selon une méthode probabiliste (roulette, tournoi, etc.), un individu pour qu'il participe à l'étape de reproduction pour former la nouvelle population.

#### – Croisement

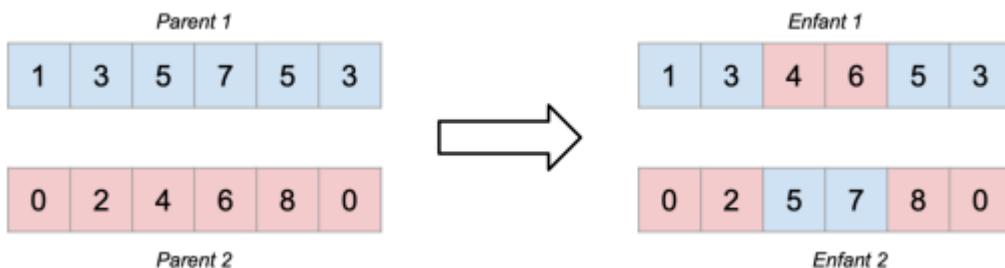
Le principal opérateur agissant sur la population des parents est le croisement, qui est appliqué avec une certaine probabilité appelée taux ou probabilité de croisement  $P_c$  (typiquement proche de l'unité). Le croisement consiste à choisir deux individus tirés au hasard dans la population courante, et à définir aléatoirement un ou plusieurs points de croisement.

Les nouveaux individus sont alors créés en interchangeant les différentes parties de chaque chaîne. Cet opérateur permet de bien exploiter le domaine de variation des individus, et de diriger la recherche vers des régions intéressantes de l'espace d'étude en utilisant la connaissance déjà présente dans la population courante.

- ✓ Croisement en un seul point :



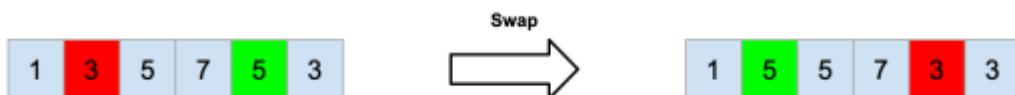
- ✓ Croisement en 2 points plus efficace :



## -Mutation

L'opération de mutation protège les algorithmes génétiques des pertes prématuées d'informations pertinentes. Elle permet d'introduire une certaine information dans la population, qui a pu être perdue lors de l'opération de croisement. Ainsi elle participe au maintien de la diversité, utile à une bonne exploration du domaine de recherche. L'opérateur de mutation s'applique avec une certaine probabilité, appelée taux ou probabilité de mutation  $P_m$ , typiquement faible.

On échange la position du 3 et du 5



## 5. Exemple d'explication :

Supposons que nous avons un VRP avec 10 clients (identifiés par les nombres de 1 à 10) et un entrepôt (identifié par 0) qui doit desservir tous les clients avec un certain nombre de véhicules.

1. **Codage des solutions** : Chaque solution dans le VRP peut être représentée sous forme de séquence de villes visitées par un véhicule. Dans ce cas, chaque séquence représentera un itinéraire pour un véhicule, et les séquences seront combinées pour former une solution complète. Nous pouvons représenter chaque séquence sous forme de liste de nombres, où chaque nombre représente l'identifiant de la ville.
2. **Génération de la population initiale** : Nous pouvons générer une population initiale aléatoirement en choisissant une séquence de villes pour chaque véhicule. Pour simplifier, supposons que nous avons deux véhicules disponibles.

Par exemple, une solution initiale pourrait être :

**Véhicule 1 :** [0, 3, 1, 7, 0]

**Véhicule 2 :** [0, 9, 6, 8, 4, 2, 5, 10, 0]

3. **Évaluation de la population** : Chaque solution est évaluée en utilisant une fonction objective qui mesure la qualité de la solution. Dans le VRP, la fonction objective pourrait être **le coût total** des trajets effectués par les véhicules. Dans ce cas, nous pouvons utiliser la **distance euclidienne** comme mesure de coût.
4. **Sélection des parents** : Nous pouvons utiliser la méthode de la roulette pour sélectionner les parents à partir de la population existante. Les parents sont sélectionnés en proportion de leur qualité (c'est-à-dire leur valeur de fonction objectif).
5. **Croisement** : Nous pouvons utiliser le croisement en deux points pour échanger une partie des séquences des parents et créer de nouvelles séquences pour les enfants. Par exemple, supposons que nous avons les deux parents suivants :

**Parent 1 :** [0, 3, 1, 7, 0]

**Parent 2 :** [0, 9, 6, 8, 4, 2, 5, 10, 0]

Nous pouvons choisir deux points de croisement aléatoires (par exemple, 2 et 5) et échanger les sous-séquences entre ces points pour créer deux enfants :

**Enfant 1 :** [0, 3, 6, 8, 4, 2, 5, 10, 0]

**Enfant 2 :** [0, 9, 1, 7, 0]

6. **Mutation :** Nous pouvons ajouter une petite perturbation aléatoire à une séquence de villes pour obtenir une nouvelle solution. Par exemple, nous pouvons choisir une solution au hasard et intervertir deux villes pour créer une nouvelle solution.
7. **Évaluation de la nouvelle population :** Chaque solution est évaluée à nouveau en utilisant la fonction objective.
8. **Répéter :** Les étapes 4 à 7 sont répétées jusqu'à ce qu'une solution satisfaisante soit trouvée ou que le nombre maximal d'itérations soit atteint.
9. **Sélection de la meilleure solution :** À la fin de l'algorithme, la meilleure solution est sélectionnée à partir de la dernière population en utilisant la fonction objective.

Ceci est juste un exemple simple pour montrer comment l'algorithme génétique peut être utilisé pour résoudre le problème du VRP. Dans la pratique, il y a de nombreux paramètres à ajuster, tels que la taille de la population, la méthode de sélection des parents et la probabilité de mutation.

## 6. Conclusion :

Dans ce chapitre, nous avons examiné trois algorithmes couramment utilisés pour résoudre les problèmes de routage des véhicules : l'algorithme Branch and Bound, l'algorithme du Plus Proche Voisin et l'algorithme Génétique. Chaque algorithme a ses avantages et ses limitations, et le choix de l'algorithme approprié dépendra du contexte spécifique du problème, des contraintes de temps, des ressources disponibles et des objectifs de l'application.

L'algorithme Branch and Bound est une approche exacte qui garantit l'optimalité de la solution, mais peut être coûteux en termes de temps de calcul pour les problèmes de grande taille. L'algorithme du Plus Proche Voisin est rapide à implémenter et donne souvent des résultats acceptables en termes de qualité de solution, mais peut ne pas toujours aboutir à la meilleure solution possible. L'algorithme Génétique offre une approche robuste pour les problèmes complexes, mais peut nécessiter plus de temps de calcul.

Il est important de considérer les avantages et les limitations de chaque algorithme lors du choix de la méthode de résolution pour un problème spécifique de routage des véhicules. De plus, les approches hybrides combinant plusieurs algorithmes peuvent également être envisagées pour tirer parti de leurs avantages respectifs. La sélection de l'algorithme approprié doit être basée sur une évaluation approfondie des exigences du problème et des contraintes du contexte d'application.

## **Partie 3 : Les solutions informatique de résolution du VRP**

## I. Introduction

Dans cette troisième partie, nous allons présenter trois solutions informatiques pour résoudre un problème VRP. La première solution est un code Python basé sur la bibliothèque Pulp, qui est une méthode de résolution exacte. La deuxième solution est un logiciel open source appelé OpenDoor Logistic Studio, qui offre des fonctionnalités pour résoudre des problèmes VRP à l'aide de méthodes heuristiques et méta heuristiques. La troisième solution est basée sur VBA d'Excel, offrant une méthode de résolution exacte pour résoudre des problèmes VRP.

## II. Solution exacte proposés Pulp :

### 1. Bibliothèques utilisées

On utilise la bibliothèque Pulp pour résoudre ce type de problème, mais on va définir quelques outils :

**Pulp** Python est une bibliothèque d'optimisation linéaire open-source pour Python. Pulp fournit une interface simple pour modéliser et résoudre des problèmes d'optimisation linéaire, en utilisant des solveurs externes tels que GLPK, CBC, CPLEX, et Gurobi.

La bibliothèque Pulp permet aux utilisateurs de définir les variables de décision, les contraintes et les objectifs de leur modèle d'optimisation linéaire de manière claire et simple. Pulp offre également des fonctionnalités pour l'importation et l'exportation de données, la définition de contraintes complexes, et la personnalisation des options de solveurs externes.

Pulp est souvent utilisé pour résoudre des problèmes d'optimisation linéaire tels que la planification de la production, l'affectation des ressources, la planification de la chaîne d'approvisionnement, et le problème du voyageur de commerce.

En résumé, Pulp Python est une bibliothèque d'optimisation linéaire pour Python qui offre une interface simple pour modéliser et résoudre des problèmes d'optimisation linéaire. Pulp est souvent utilisé pour résoudre des problèmes de planification de production, d'affectation de ressources, de planification de la chaîne d'approvisionnement, et le problème du voyageur de commerce.

**Seaborn** est une bibliothèque Python open-source de visualisation de données basée sur Matplotlib. Seaborn fournit des fonctionnalités pour créer des graphiques statistiques attrayants et informatifs, avec un code minimal.

Seaborn est souvent utilisé pour explorer des données et trouver des tendances, des relations et des modèles cachés. Il offre des graphiques tels que les histogrammes, les nuages de points, les diagrammes en boîte, les graphiques à barres, les diagrammes de dispersion, les diagrammes de violon, les cartes thermiques, et bien plus encore. Seaborn fournit également des fonctionnalités pour ajuster les paramètres de graphique, comme les couleurs, les thèmes, les styles et les palettes.

**Matplotlib** est une bibliothèque de visualisation de données en Python, open-source et très populaire. Elle fournit des fonctionnalités pour créer des graphiques en deux dimensions (2D) et en trois dimensions (3D), des animations, des tableaux de bord, etc. Matplotlib est très flexible et offre de nombreux paramètres pour personnaliser les graphiques, y compris les couleurs, les étiquettes d'axes, les annotations, les styles de ligne, etc.

**Matplotlib** permet de créer de nombreux types de graphiques, tels que les graphiques en barres, les graphiques en nuage de points, les graphiques à aires, les diagrammes de dispersion, les graphiques en boîtes, les graphiques en violons, les diagrammes de Gantt, les cartes thermiques, les diagrammes de Sankey, etc.

**NumPy** est une bibliothèque Python open-source destinée à la manipulation de tableaux multidimensionnels et de matrices, ainsi qu'à la réalisation de calculs mathématiques sur ces tableaux. NumPy fournit une grande variété de fonctions mathématiques pour travailler avec des tableaux, comme les fonctions trigonométriques, les fonctions exponentielles et logarithmiques, les fonctions de statistiques, etc.

**NumPy** est souvent utilisé pour le calcul numérique en science des données, en apprentissage automatique et en traitement du signal. Cette bibliothèque permet également de réaliser des opérations de traitement d'images et de sons.

**NumPy** est également utilisé en conjonction avec d'autres bibliothèques Python pour la science des données, telles que Pandas, SciPy, Scikit-learn, Matplotlib, etc.

## 2. Implantation :

Puisque le problème est difficile, on va regrouper les provinces sous 5 régions pour réduire la complexité du problème et on va résoudre les 5 sous-problèmes selon les provinces de distances proches.

### Region 1:

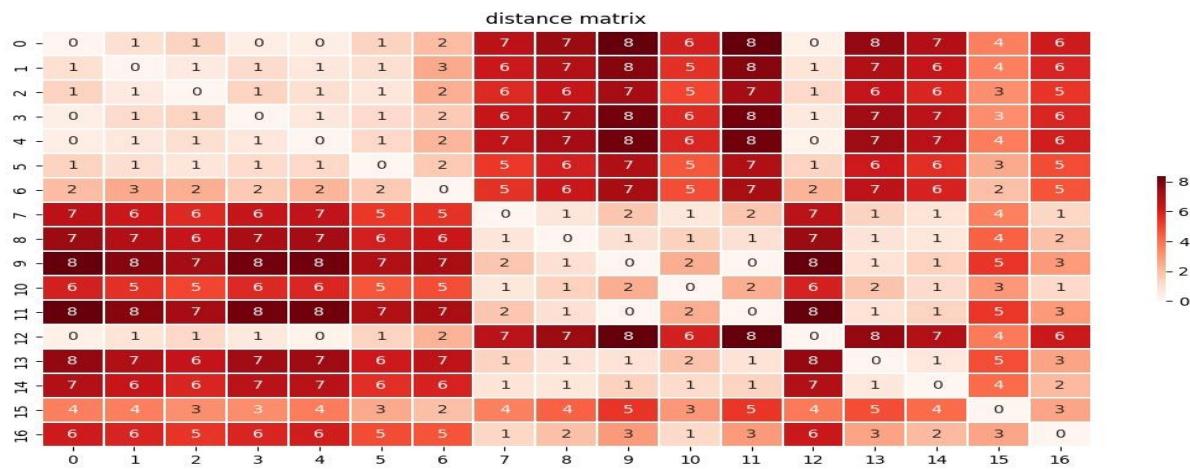


Figure 1 Matrice des distances région 1

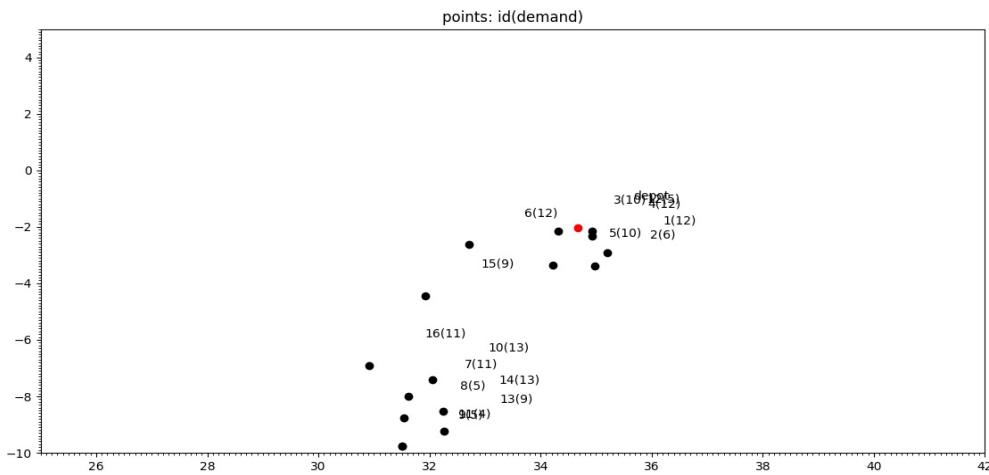


Figure 2 représentation des clients et dépôt région 1

|    | province          | x         | y         | demand |
|----|-------------------|-----------|-----------|--------|
| 0  | Oujda-Angad       | 34.668328 | -2.036245 | 0      |
| 1  | Nador             | 35.207006 | -2.912547 | 12     |
| 2  | Driouch           | 34.976940 | -3.391005 | 6      |
| 3  | Jerada            | 34.310531 | -2.159979 | 10     |
| 4  | Berkan            | 34.926676 | -2.329409 | 12     |
| 5  | Guercif           | 34.225576 | -3.352345 | 10     |
| 6  | Figuig            | 32.705884 | -2.628766 | 12     |
| 7  | Marrakech         | 31.625826 | -7.989161 | 11     |
| 8  | Chichaoua         | 31.546903 | -8.759546 | 5      |
| 9  | Al Haouz          | 31.512927 | -9.767879 | 5      |
| 10 | Kelâa des Sraghnâ | 32.054330 | -7.406760 | 13     |
| 11 | Essaouira         | 31.511828 | -9.762090 | 4      |
| 12 | Rehamna           | 34.921334 | -2.146031 | 5      |
| 13 | Safi              | 32.265078 | -9.230554 | 9      |
| 14 | Youssoufia        | 32.245801 | -8.532439 | 13     |
| 15 | Errachidia        | 31.929089 | -4.434081 | 9      |
| 16 | Ouarzazate        | 30.920193 | -6.910923 | 11     |

Figure 3 Cordonnées des provinces de région 1

### Solution Region 1:

Result - Optimal solution found

Objective value: 31.49127526  
 Enumerated nodes: 0  
 Total iterations: 0  
 Time (CPU seconds): 21.64  
 Time (Wallclock seconds): 21.64

Option for printingOptions changed from normal to all  
 Total time (CPU seconds): 25.87 (Wallclock seconds): 25.87

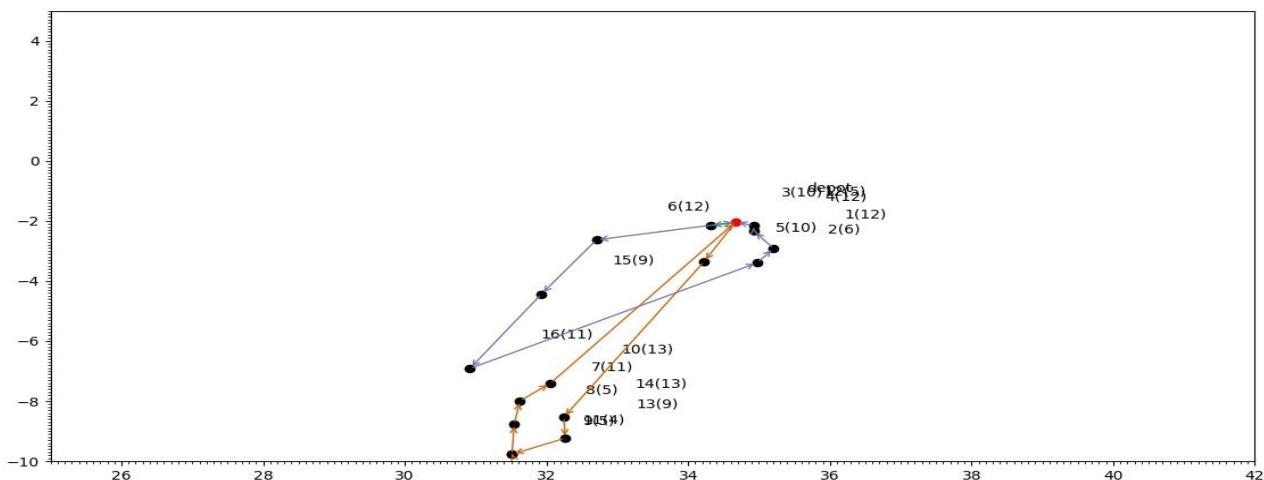


Figure 4 Représentation de la solution de région 1

## Region 2:

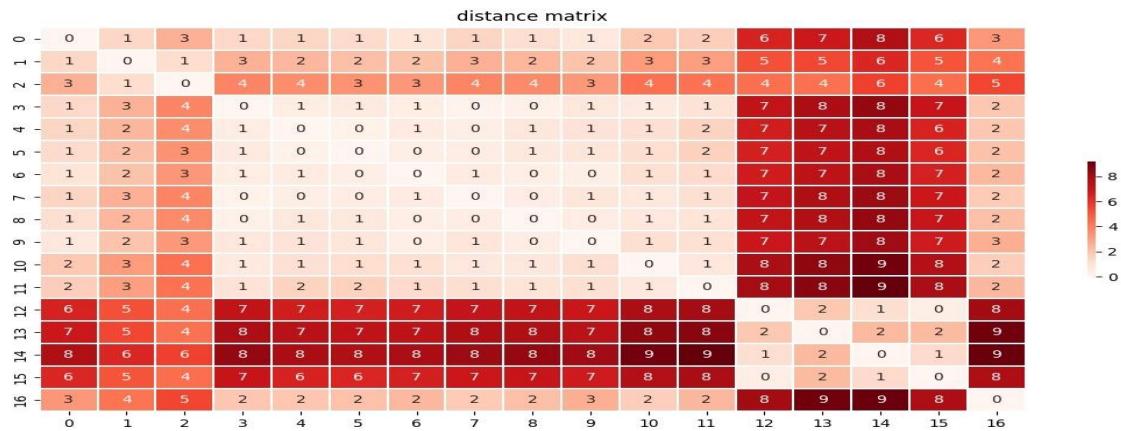


Figure 5 matrice des distances de région 2

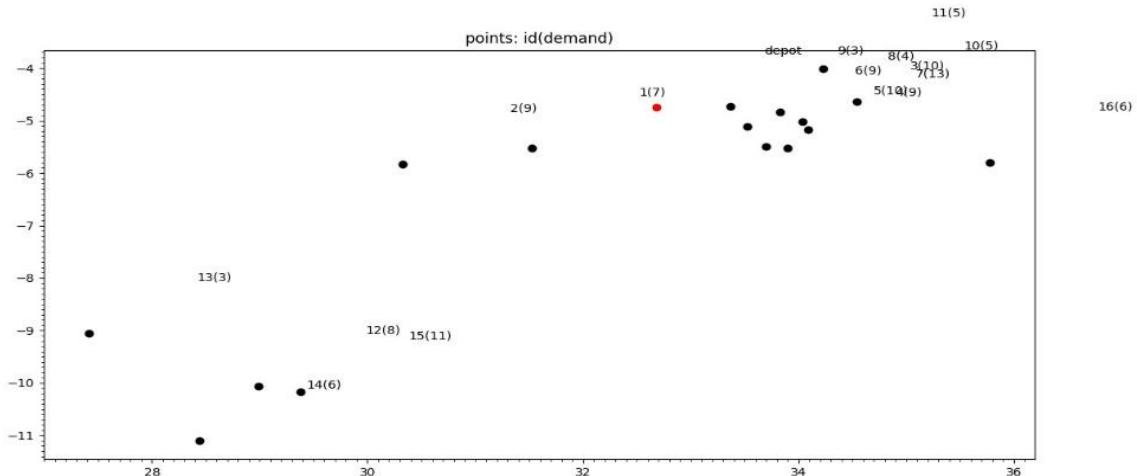


Figure 6 représentation des clients et dépôt région 2

Solution Region2 :

Objective value: 25.71147028

Enumerated nodes: 0

Total iterations: 9

Time (CPU seconds): 192.92

Time (Wallclock seconds): 192.92

Option for printingOptions changed from normal to all

Total time (CPU seconds): 196.86 (Wallclock seconds): 196.86

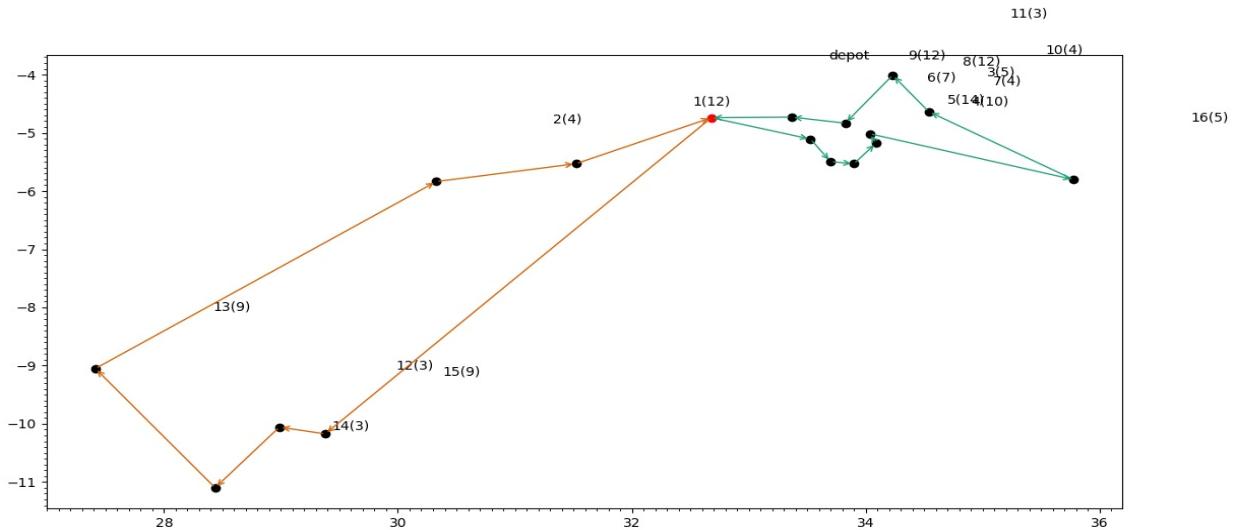


Figure 7 Représentation de la solution de région 2

### Region 3 :

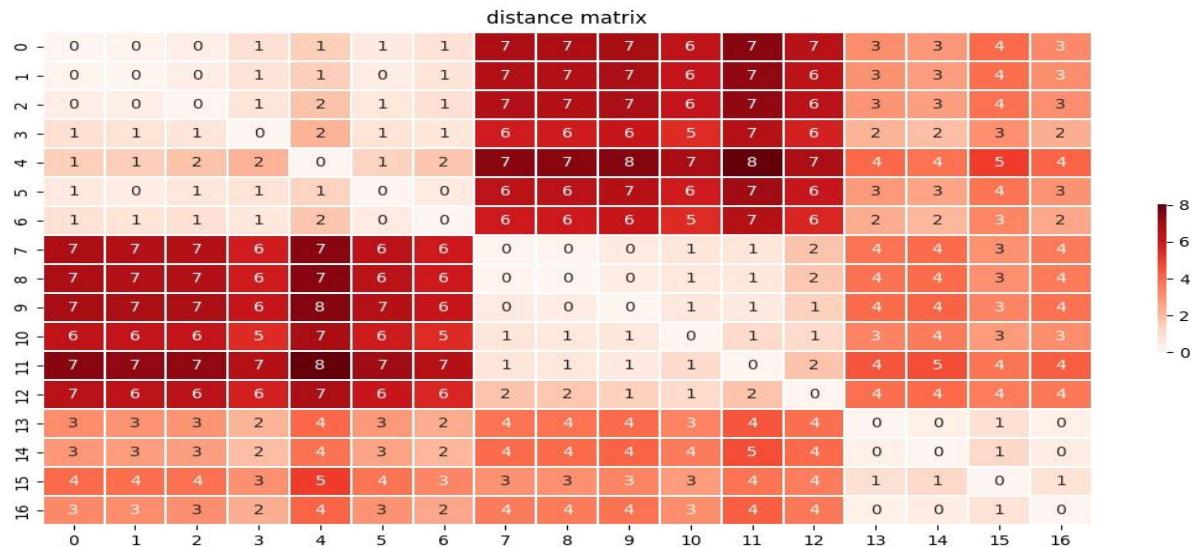


Figure 8 matrice des distances de région 3

|    | province             | x         | y         | demand |
|----|----------------------|-----------|-----------|--------|
| 0  | M'diq-Fnideq         | 35.683360 | -5.323216 | 0      |
| 1  | Tétouan              | 35.570175 | -5.374278 | 4      |
| 2  | Fahs-Anjra           | 35.758900 | -5.652058 | 13     |
| 3  | Larache              | 35.195233 | -6.152913 | 12     |
| 4  | Al Hoceima           | 35.245114 | -3.930186 | 10     |
| 5  | Chefchaouen          | 35.170083 | -5.276658 | 9      |
| 6  | Ouazzane             | 34.796757 | -5.578493 | 4      |
| 7  | Agadir Ida-Ou-Tanane | 30.420949 | -9.604430 | 14     |
| 8  | Inezgane-Aït Melloul | 30.356293 | -9.545935 | 14     |
| 9  | Chtouka-Aït Baha     | 30.008060 | -9.333441 | 10     |
| 10 | Taroudannt           | 30.471943 | -8.866928 | 3      |
| 11 | Tiznit               | 29.698624 | -9.731281 | 14     |
| 12 | Tata                 | 29.746392 | -7.969605 | 3      |
| 13 | Casablanca           | 33.594514 | -7.620028 | 12     |
| 14 | Mohammedia           | 33.695838 | -7.389329 | 7      |
| 15 | El Jadida            | 33.233695 | -8.500425 | 7      |
| 16 | Nouaceur             | 33.379465 | -7.618914 | 5      |

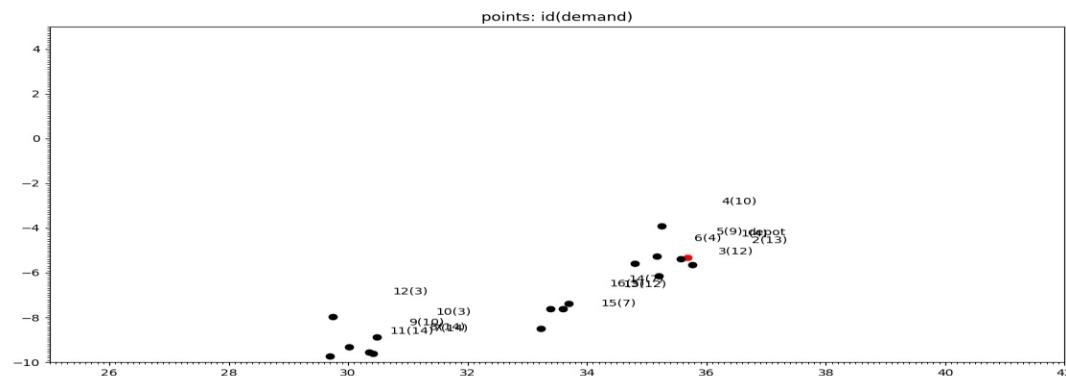


Figure 9 représentation des clients et dépôt région 3

#### Solution Région 4:

Result - Optimal solution found

```
Objective value:          20.35869388
Enumerated nodes:         0
Total iterations:          0
Time (CPU seconds):       19.36
Time (Wallclock seconds): 19.36
```

```
Option for printingOptions changed from normal to all
Total time (CPU seconds): 23.18   (Wallclock seconds): 23.18
```

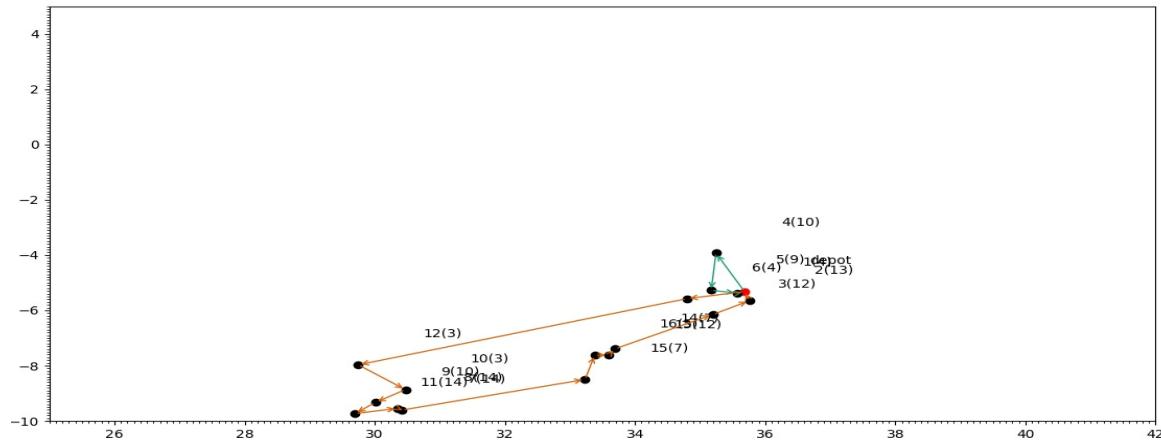


Figure 10 Représentation de la solution de région 3.

#### Région 4:

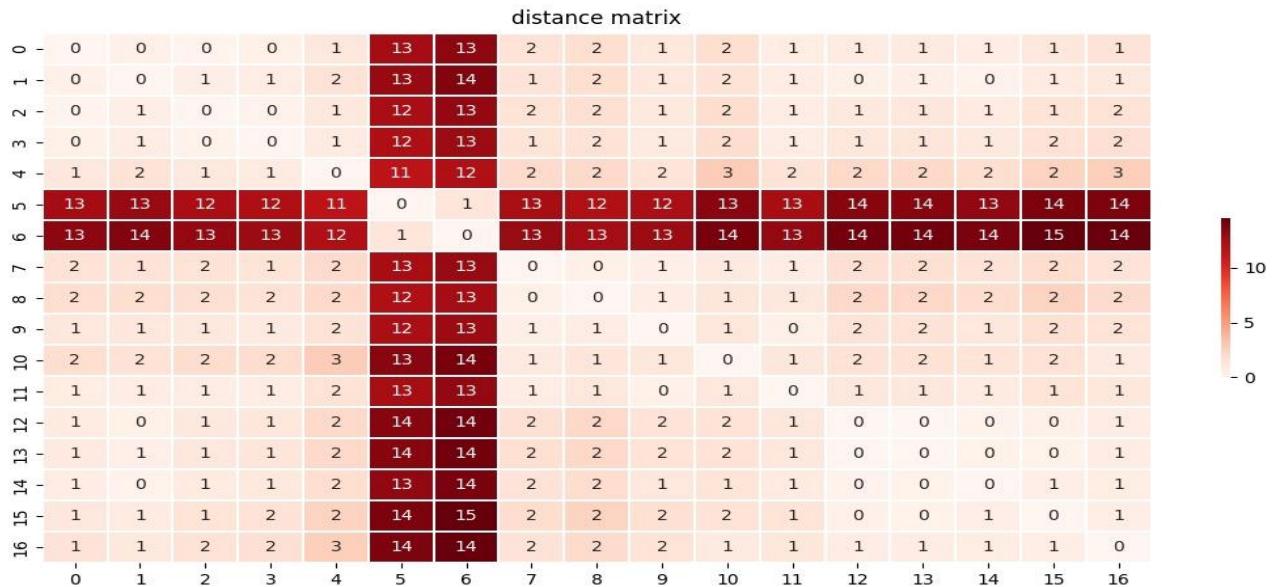


Figure 11 matrice des distances de région 3

|    | province        | x         | y          | demand |
|----|-----------------|-----------|------------|--------|
| 0  | Médiouna        | 33.456251 | -7.519440  | 0      |
| 1  | Benslimane      | 33.616871 | -7.123036  | 5      |
| 2  | Berrechid       | 33.267675 | -7.581147  | 7      |
| 3  | Settat          | 33.002397 | -7.619867  | 9      |
| 4  | Sidi Bennour    | 32.650779 | -8.424209  | 5      |
| 5  | Oued Ed-Dahab   | 23.283710 | -15.020262 | 8      |
| 6  | Aousserd        | 22.072415 | -14.445243 | 7      |
| 7  | Béni Mellal     | 32.334193 | -6.353335  | 13     |
| 8  | Azilal          | 31.870226 | -6.432248  | 9      |
| 9  | Equih Ben Salah | 32.509832 | -6.854521  | 7      |
| 10 | Khénifra        | 32.935772 | -5.669650  | 14     |
| 11 | Khouribga       | 32.885508 | -6.909238  | 10     |
| 12 | Rabat           | 34.022360 | -6.834022  | 10     |
| 13 | Salé            | 34.044889 | -6.814017  | 7      |
| 14 | Skhirate-Témara | 33.747930 | -6.879220  | 8      |
| 15 | Kénitra         | 34.264570 | -6.570169  | 9      |
| 16 | Khémisset       | 33.830287 | -6.072605  | 4      |

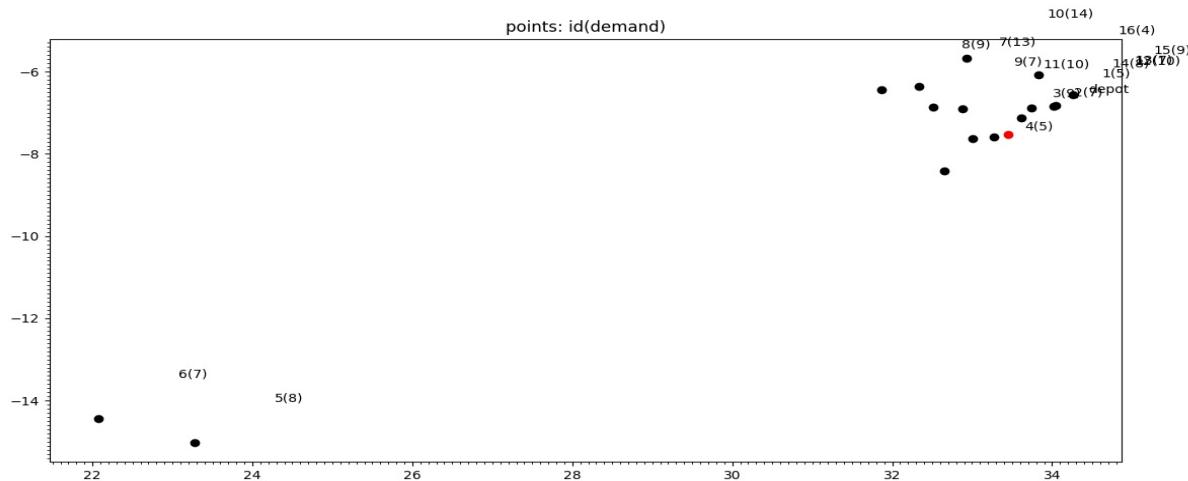


Figure 12 représentation des clients et dépôt région 3

#### Solution Région 4:

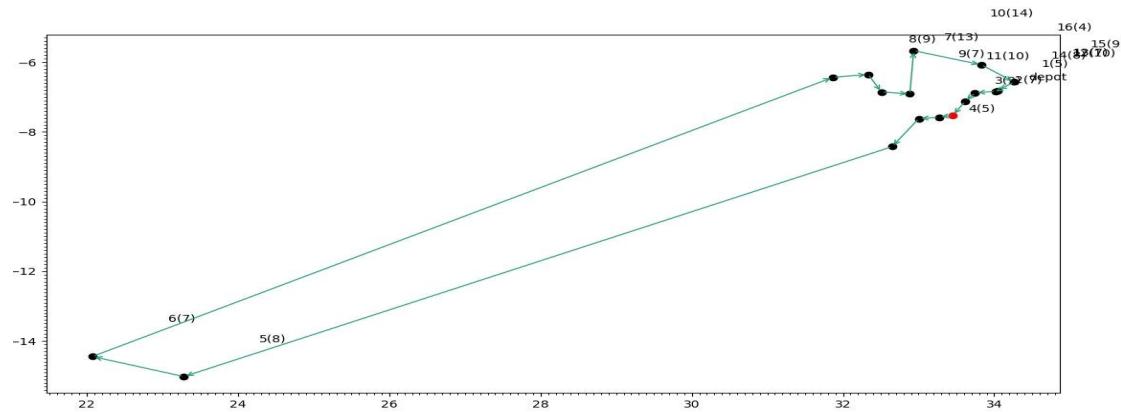


Figure 13 Représentation de la solution de région 3.

Result - Optimal solution found

Objective value: 32.40329831  
Enumerated nodes: 0  
Total iterations: 0  
Time (CPU seconds): 22.61  
Time (Wallclock seconds): 22.61

Option for printingOptions changed from normal to all  
Total time (CPU seconds): 28.28 (Wallclock seconds): 28.28

## Région 5:

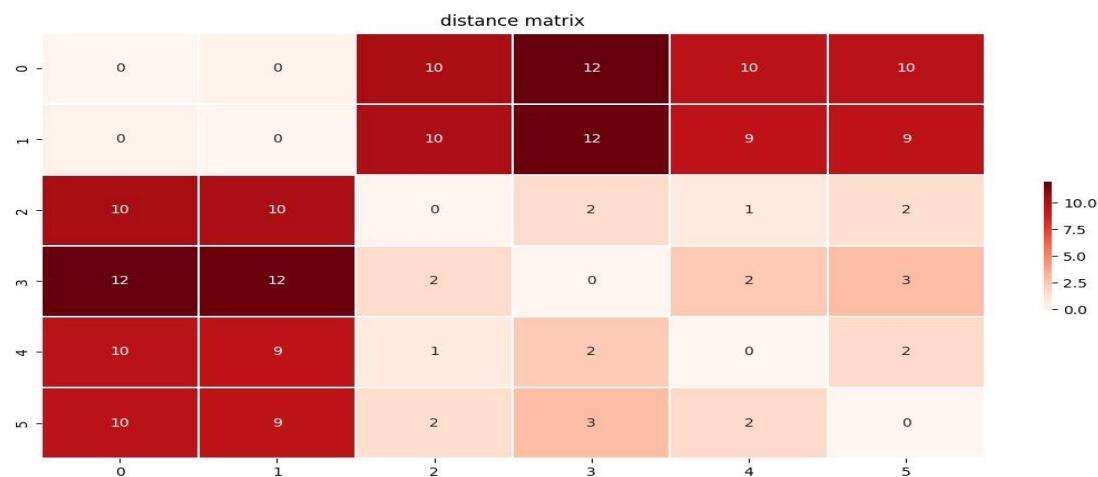


Figure 14 matrice des distances de région 3

|   | province     | x         | y          | demand |
|---|--------------|-----------|------------|--------|
| 0 | Sidi Kacem   | 34.226412 | -5.711434  | 0      |
| 1 | Sidi Slimane | 34.259878 | -5.927253  | 10     |
| 2 | Laâyoune     | 27.154512 | -13.195392 | 6      |
| 3 | Boujdour     | 26.126338 | -14.483654 | 7      |
| 4 | Tarfaya      | 27.940120 | -12.924993 | 12     |
| 5 | Es-Semara    | 26.740000 | -11.670000 | 9      |

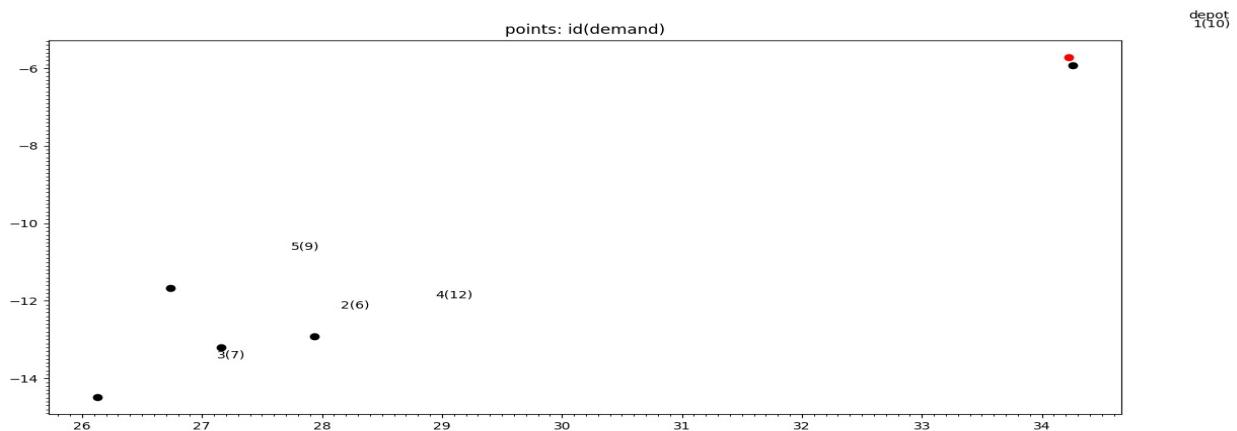


Figure 15 représentation des clients et dépôt région 3

### Solution Région 5:

Result - Optimal solution found

|                           |             |
|---------------------------|-------------|
| Objective value:          | 24.57460556 |
| Enumerated nodes:         | 0           |
| Total iterations:         | 0           |
| Time (CPU seconds):       | 0.01        |
| Time (Wallclock seconds): | 0.01        |

```
Option for printingOptions changed from normal to all
Total time (CPU seconds):      0.03   (Wallclock seconds):      0.03
```

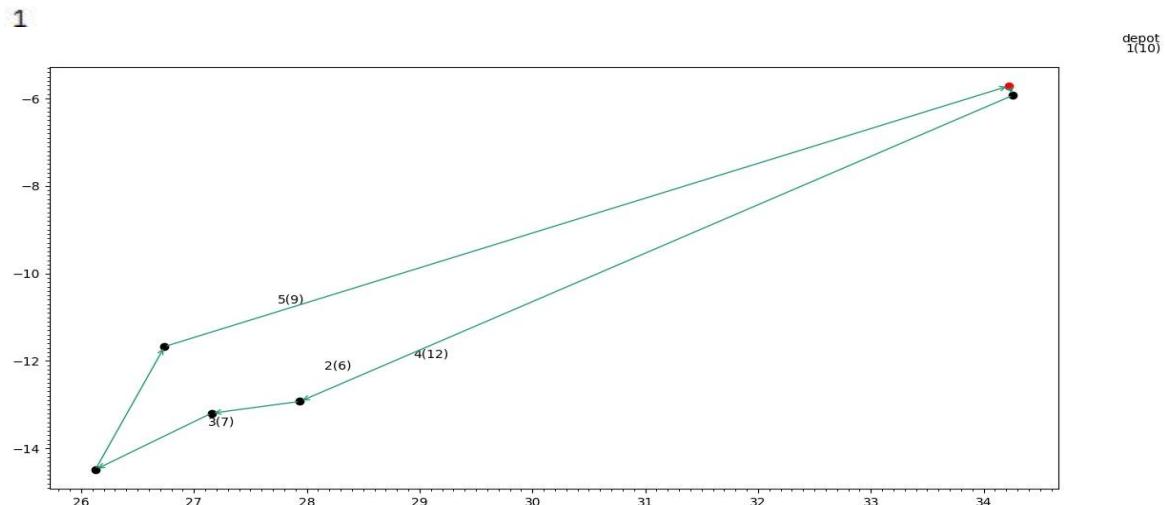


Figure 16 Représentation de la solution de région 3.

### III. Solution informatique en utilisant l'outil : OpenDoorLogistics Studio

#### 1. Présentation de OpendoorLogistic Studio :

OpenDoorLogistics Studio est un logiciel de planification de tournées de véhicules (VRP) développé par la société OpenDoorLogistics Ltd. Il est conçu pour aider les entreprises à optimiser leurs opérations de livraison en minimisant les coûts et en maximisant l'efficacité.

L'outil utilise des algorithmes avancés pour résoudre des problèmes de VRP complexes, tels que l'optimisation des itinéraires, la planification des horaires de livraison, la prise en compte de divers types de contraintes, la modélisation de la capacité des véhicules et la modélisation des coûts de transport. OpenDoorLogistics Studio permet également aux utilisateurs de personnaliser la modélisation du problème en fonction de leurs besoins spécifiques.

Il est disponible en version d'essai gratuite ainsi qu'en version payante avec des fonctionnalités supplémentaires et un support technique.

Les données sont importées depuis un fichier Excel, ce qui permet de générer des cartes de territoires optimisées en fonction de la demande et d'autres facteurs. Les entreprises peuvent ainsi minimiser les coûts de leurs opérations de vente ou de livraison tout en améliorant la satisfaction des clients. Les données peuvent être modifiées à la fois dans Open Door Logistics Studio et dans Excel, ce qui permet de travailler facilement entre les deux applications.

#### 2. Raisons du choix de OpenDoorLogistics Studio :

On a choisi d'utiliser OpenDoorLogistics Studio pour plusieurs raisons ; voici quelques-unes pour lesquelles cet outil s'avère être un choix judicieux pour résoudre le VRP :

- **Fonctionnalités avancées :** OpenDoorLogistics Studio offre des fonctionnalités avancées pour la résolution de problèmes de VRP, telles que la prise en compte de plusieurs types de contraintes, la prise en compte de plusieurs modes de transport, la modélisation de la capacité des véhicules et la modélisation des coûts de transport.

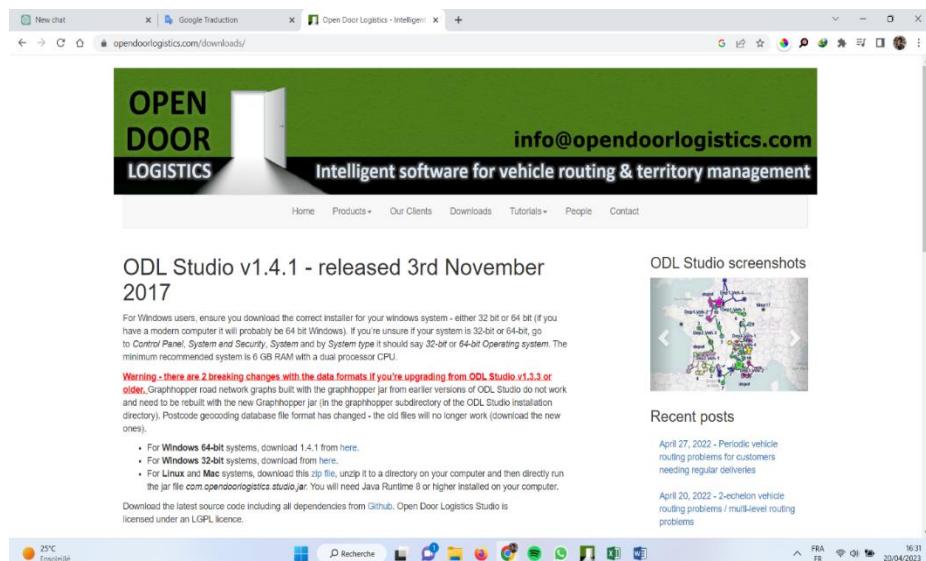
- **Flexibilité :** L'outil est très flexible et permet de personnaliser la modélisation du problème en fonction des besoins spécifiques de l'utilisateur. Par exemple, l'utilisateur peut définir ses propres contraintes et objectifs.
- **Facilité d'utilisation :** OpenDoorLogistics Studio est relativement facile à utiliser grâce à son interface utilisateur conviviale et à ses guides d'utilisation détaillés. L'outil prend également en charge l'importation de données à partir de divers formats de fichiers.
- **Performance :** OpenDoorLogistics Studio est conçu pour résoudre rapidement des problèmes de VRP de grande taille. L'outil utilise des algorithmes avancés pour optimiser les itinéraires et les horaires de livraison, ce qui permet de réaliser des économies importantes en termes de coûts et de temps.

### 3. Résolution :

Pour résoudre un problème de tournés de véhicules utilisant opendoor logistic studio, il faut suivre les étapes suivantes :

#### Téléchargement et installation :

Le logiciel est disponible sur le site web <https://www.opendoorlogistics.com/downloads/>

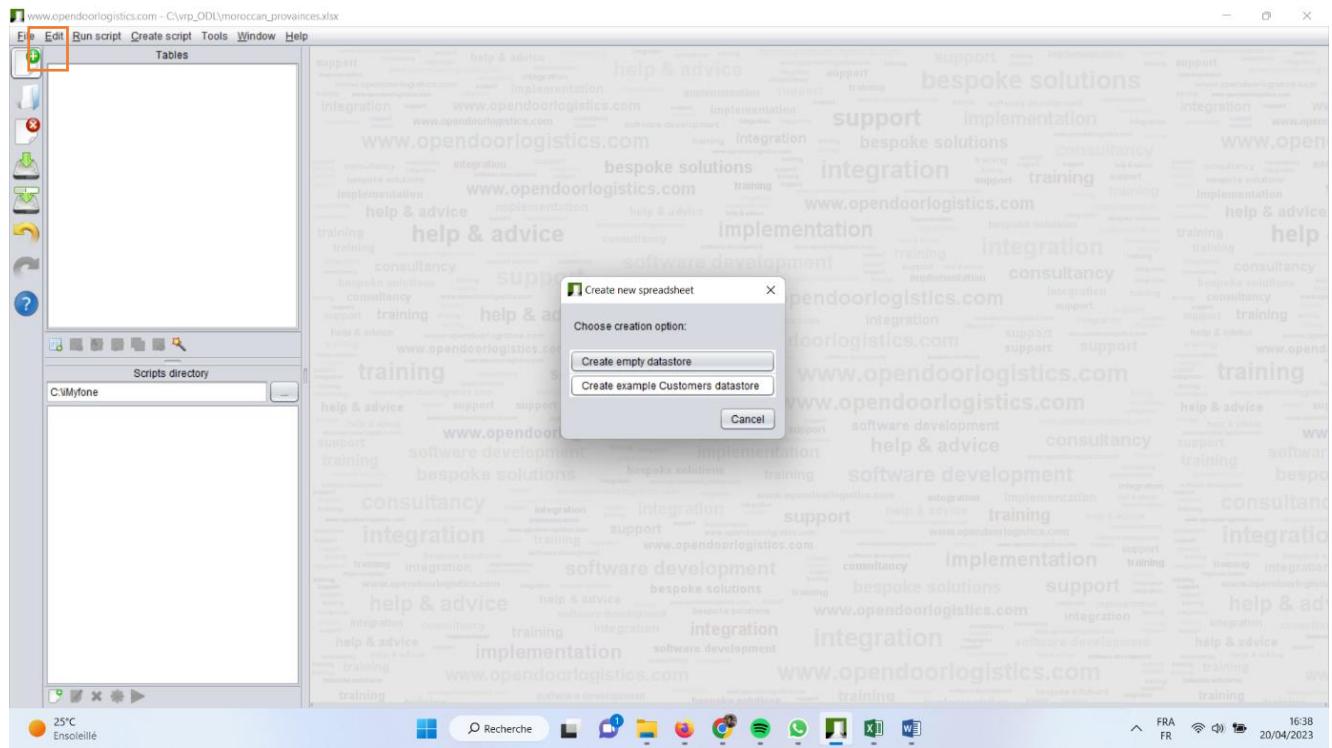


**Figure 17 : Page de téléchargement d'opendoorlogistics Studio**

Après le téléchargement, On lance l'installation qui est simple et claire.

## **Création de new file :**

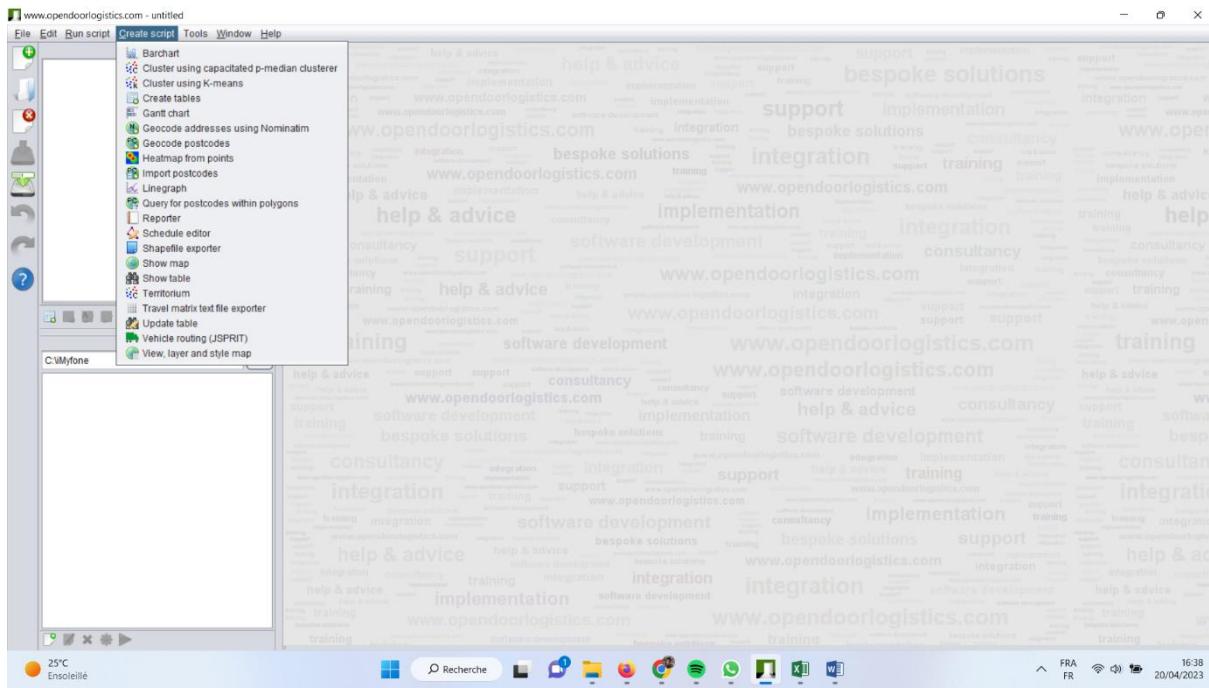
On crée un nouveau fichier vide en cliquant sur le bouton indiquer dans l'image ci-dessous



### **Figure 18 Creation de nouveau fichier**

## **Créer un Script :**

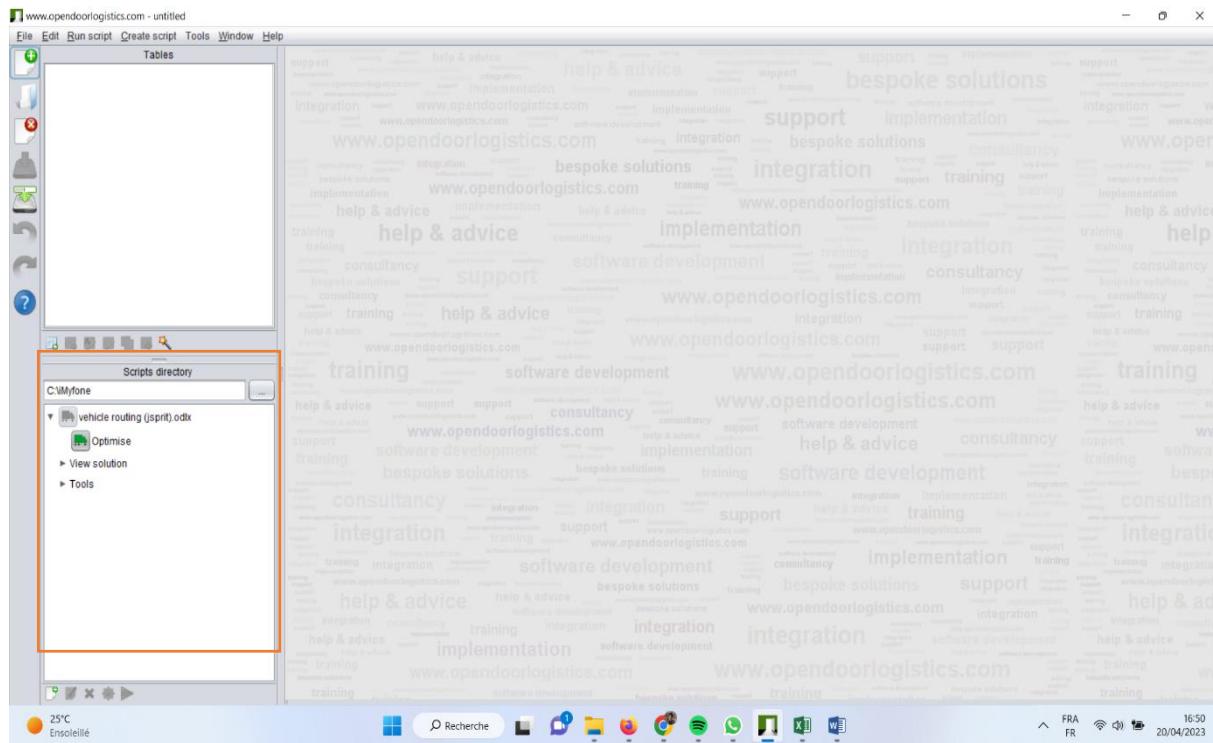
Après la création de fichier sur opendoorlogistics Studio, maintenant on va créer un nouveau script d'optimisation de VRP, en cliquant sur « Create Script » et on choisit « Vehicle routing (JSRIP) ». Puis on sauvegarde le fichier dans un répertoire donné.



Après la sauvegarde

**Figure 4 :Créer un script**

l'ensemble d'instruction dont on a besoin durant notre résolution de VRP, c'est un script Java qui s'exécute derrière le clique de chaque item.



**Figure 19 :menu d'instructions utiles**

## **Création des tableaux de données :**

Les inputs de logiciel sont des fichiers Excel, qui contiennent les informations sur les clients, les dépôts, les capacités etc. On va discuter en détails ces tables juste après. Pour créer ces tables, rendez-vous sur la partie Script en cliquant sur « tools » puis « create all inputs tables » :

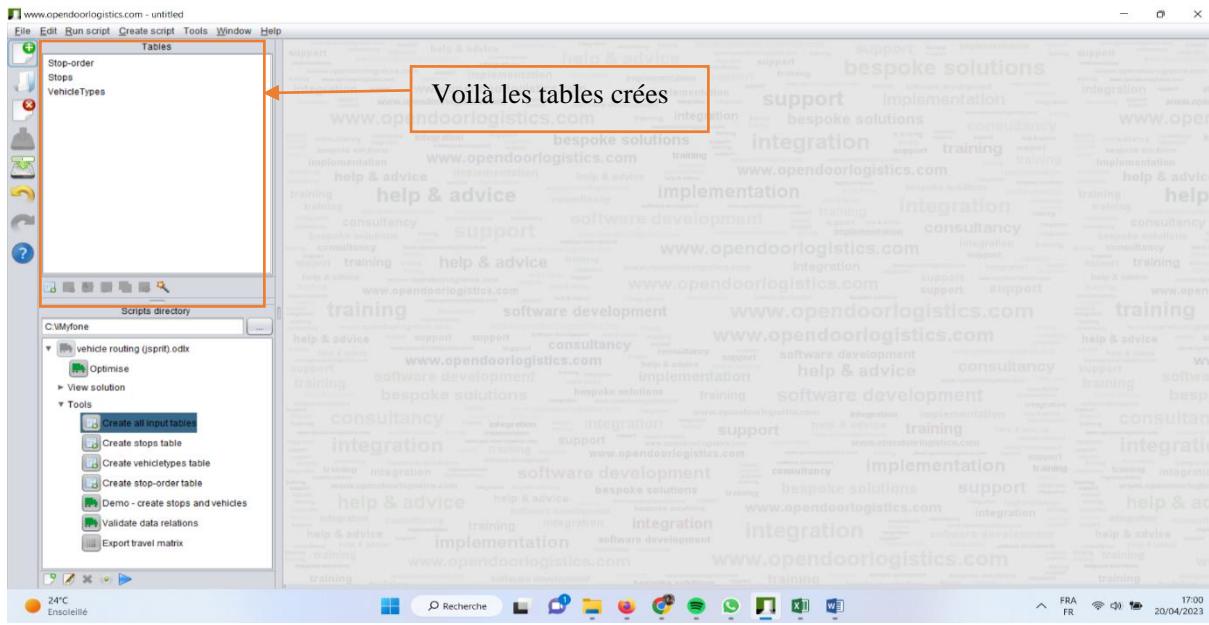


Figure 20 Création des tableaux de données

- ✓ La table Stop-order : contient l'ordre des tournés fait par les véhicules
- ✓ La table Stops : contient les informations sur les clients, leurs demandes, leurs adresses, les coordonnées etc...
- ✓ Vehicle Type : contient les informations sur chaque dépôt.

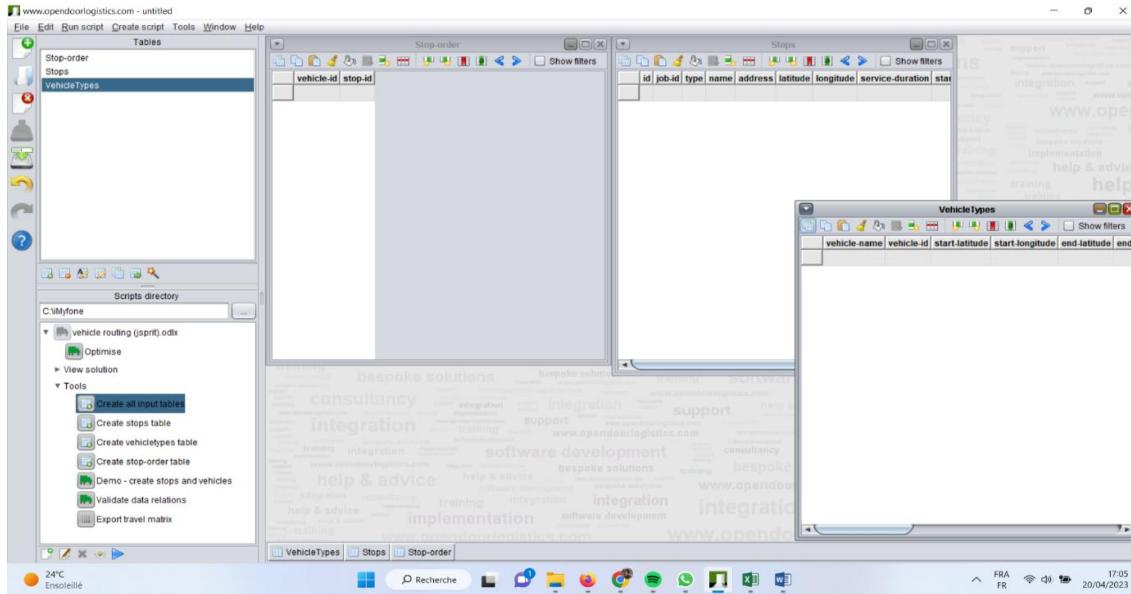


Figure 21 :Inputs Tables

## Les données utilisées :

Dans notre cas on souhaite réaliser une optimisation des tournées ayant à disposition 20 véhicules, les clients sont dans chaque province de Maroc, le dépôt est à Béni Mellal. Pour cela nous avons besoin des coordonnées de chaque province, on a utilisé une extension de google sheet pour avoir les géocode de chaque province.

|    | A                 | B          | C                  | D | E | F | G | H | I | J |
|----|-------------------|------------|--------------------|---|---|---|---|---|---|---|
| 1  | Full Address      | Latitude   | Longitude          |   |   |   |   |   |   |   |
| 2  | Oujda-Angad       | 34.6790031 | -1.8905963         |   |   |   |   |   |   |   |
| 3  | Nador             | 35.1686165 | -2.9275836         |   |   |   |   |   |   |   |
| 4  | Driouch           | 34.9792528 | -3.398304          |   |   |   |   |   |   |   |
| 5  | Jerada            | 34.3061791 | -2.1794136         |   |   |   |   |   |   |   |
| 6  | Berkana,Morocco   | 34.9177281 | -2.3123031         |   |   |   |   |   |   |   |
| 7  | Taourirt          | 34.3983716 | -2.8935028         |   |   |   |   |   |   |   |
| 8  | Guercif           | 34.2299391 | -3.3489671         |   |   |   |   |   |   |   |
| 9  | Figuig            | 32.1092613 | -1.229806          |   |   |   |   |   |   |   |
| 10 | Marrakech         | 31.6294723 | -7.981084500000001 |   |   |   |   |   |   |   |
| 11 | Chichaoua         | 31.5383581 | -8.7646388         |   |   |   |   |   |   |   |
| 12 | Al Haouz          | 31.2956729 | -7.872159999999999 |   |   |   |   |   |   |   |
| 13 | Kelâa des Sraghnâ | 32.0550961 | -7.400287199999999 |   |   |   |   |   |   |   |
| 14 | Essaouira         | 31.5084926 | -9.7595041         |   |   |   |   |   |   |   |
| 15 | Rehamna           | 32.286308  | -7.959286299999999 |   |   |   |   |   |   |   |
| 16 | Safi, Morocco     | 32.3008151 | -9.2272033         |   |   |   |   |   |   |   |
| 17 | Youssoufia        | 32.2438426 | -8.5226706         |   |   |   |   |   |   |   |
| 18 | Errachidia        | 31.927236  | -4.4284985         |   |   |   |   |   |   |   |
| 19 | Ouarzazate        | 30.9335436 | -6.937016          |   |   |   |   |   |   |   |
| 20 | Midelt            | 32.6799423 | -4.7329268         |   |   |   |   |   |   |   |

Figure 22 Génération des géocode de chaque province utilisant Google sheet

Après on intègre ces informations dans le fichier Excel input.

|    | B      | C                 | D                 | E          | F                  | G         | H                | I          | J        | K        | L               | M |
|----|--------|-------------------|-------------------|------------|--------------------|-----------|------------------|------------|----------|----------|-----------------|---|
| 1  | job-id | type              | name              | address    | latitude           | longitude | service-duration | start-time | end-time | quantity | required-skills |   |
| 2  | D      | Oujda-Angad       | Oujda-Angad       | 34.6790031 | -1.8905963         |           |                  |            |          | 297      |                 |   |
| 3  | D      | Nador             | Nador             | 35.1686165 | -2.9275836         |           |                  |            |          | 230      |                 |   |
| 4  | D      | Driouch           | Driouch           | 34.9792528 | -3.398304          |           |                  |            |          | 281      |                 |   |
| 5  | D      | Jerada            | Jerada            | 34.3061791 | -2.1794136         |           |                  |            |          | 260      |                 |   |
| 6  | D      | Berkana,Morocco   | Berkana,Morocco   | 34.9177281 | -2.3123031         |           |                  |            |          | 300      |                 |   |
| 7  | D      | Taourirt          | Taourirt          | 34.3983716 | -2.8935028         |           |                  |            |          | 361      |                 |   |
| 8  | D      | Guercif           | Guercif           | 34.2299391 | -3.3489671         |           |                  |            |          | 84       |                 |   |
| 9  | D      | Figuig            | Figuig            | 32.1092613 | -1.229806          |           |                  |            |          | 112      |                 |   |
| 10 | D      | Marrakech         | Marrakech         | 31.6294723 | -7.981084500000001 |           |                  |            |          | 390      |                 |   |
| 11 | D      | Chichaoua         | Chichaoua         | 31.5383581 | -8.7646388         |           |                  |            |          | 416      |                 |   |
| 12 | D      | Al Haouz          | Al Haouz          | 31.2956729 | -7.872159999999999 |           |                  |            |          | 314      |                 |   |
| 13 | D      | Kelâa des Sraghnâ | Kelâa des Sraghnâ | 32.0550961 | -7.400287199999999 |           |                  |            |          | 229      |                 |   |
| 14 | D      | Essaouira         | Essaouira         | 31.5084926 | -9.7595041         |           |                  |            |          | 227      |                 |   |
| 15 | D      | Rehamna           | Rehamna           | 32.286308  | -7.959286299999999 |           |                  |            |          | 493      |                 |   |
| 16 | D      | Safi, Morocco     | Safi, Morocco     | 32.3008151 | -9.2272033         |           |                  |            |          | 217      |                 |   |
| 17 | D      | Youssoufia        | Youssoufia        | 32.2438426 | -8.5226706         |           |                  |            |          | 311      |                 |   |
| 18 | D      | Errachidia        | Errachidia        | 31.927236  | -4.4284985         |           |                  |            |          | 88       |                 |   |
| 19 | D      | Ouarzazate        | Ouarzazate        | 30.9335436 | -6.937016          |           |                  |            |          | 276      |                 |   |
| 20 | D      | Midelt            | Midelt            | 32.6799423 | -4.7329268         |           |                  |            |          | 69       |                 |   |

Figure 23 Fichier Input pour les provinces du Maroc

## Chargement d'input table sur ODL studio :

Pour charger le fichier qui contient nos données d'entrées on clique sur le bouton comme illustré dans l'image ci-dessous :

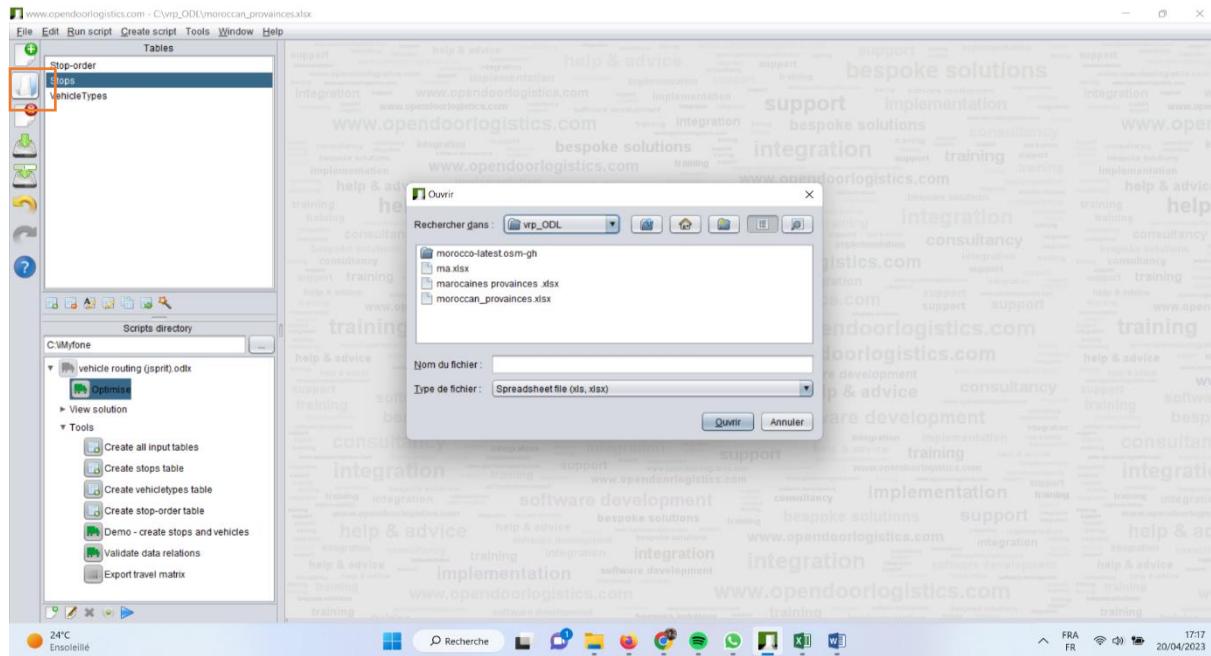


Figure 24 Chargement d'input table sur ODL studio

## Optimiser :

Après on clique sur optimiser afin de trouver la bonne solution dans une courte durée

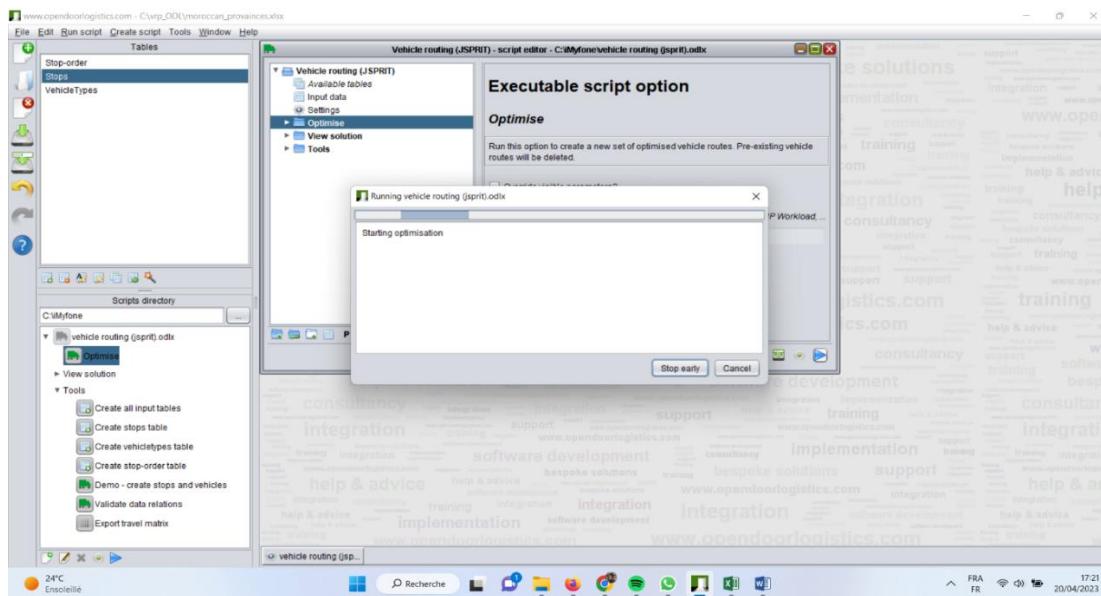


Figure 25 Optimiser

### **Visualisation de solution :**

OpenDoor Logistics utilise GraphHopper pour calculer les itinéraires pour résoudre les problèmes de VRP, puis affiche les résultats sur une carte pour visualiser les itinéraires et les emplacements des clients.

Pour ce faire on clique sur view solution qui est dans la partie script, puis on clique sur « view route in map »

On utilise la souris pour explorer toutes les routes dans la carte, on peut zoomer, éditer les routes et aussi exporter les résultats. On va voir cette partie en détails dans la présentation

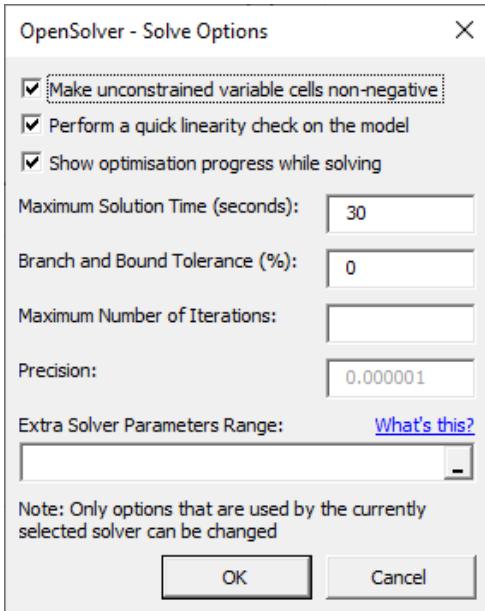
### **4. Extensions :**

OpenDoor Logistics Studio est un logiciel qui offre plusieurs extensions pour étendre ses fonctionnalités et répondre aux besoins spécifiques de ses utilisateurs. Voici quelques-unes des extensions les plus populaires :

- ✓ ODL Geocoding - Cette extension permet de géocoder des adresses pour les convertir en coordonnées géographiques, ce qui facilite la création de cartes et d'itinéraires pour les opérations de livraison.
- ✓ ODL Route Optimiser - Cette extension permet de calculer les itinéraires optimaux pour les opérations de livraison, en prenant en compte des facteurs tels que la distance, le temps de trajet, la capacité des véhicules et les préférences des clients.
- ✓ ODL Territory Planner - Cette extension permet de créer des territoires de vente ou de livraison en fonction de critères tels que la densité de la population, le chiffre d'affaires et les objectifs de couverture géographique.
- ✓ ODL Mapmaker - Cette extension permet de créer des cartes personnalisées à partir de données géographiques et de les intégrer dans les rapports et les présentations. ODL PDF
- ✓ Report Writer - Cette extension permet de générer des rapports PDF personnalisés à partir de données géographiques et de cartes, ce qui facilite la communication et la présentation des résultats.

Ces extensions permettent aux utilisateurs d'OpenDoor Logistics Studio de personnaliser le logiciel en fonction de leurs besoins spécifiques et de tirer le meilleur parti de ses fonctionnalités pour optimiser leurs opérations de livraison et de logistique.

## IV. Outil exacte --Open Solver (COIN-OR CBC)--



Le Problème de Routage de Véhicules (VRP) est un problème classique dans le domaine de la logistique et de la recherche opérationnelle. Il consiste à trouver le moyen le plus efficace de router une flotte de véhicules pour desservir un ensemble de clients avec des demandes connues, tout en minimisant la distance totale parcourue. Dans ce rapport, nous allons démontrer comment utiliser Excel VBA et Solver avec le moteur COIN-OR CBC pour résoudre un problème de PRV de 5 régions.

### 1. Énoncé du problème :

Nous disposons d'une flotte de véhicules qui doivent livrer des marchandises à des clients dans 5 régions. Chaque région a un ensemble de clients avec des demandes connues. L'objectif est de trouver le moyen le plus efficace de router les véhicules pour servir tous les clients tout en minimisant la distance totale parcourue.

### 2. Données :

- ✓ Nous disposons des données suivantes pour le problème:  
Régions: 5 Clients: Nbr Véhicules: Var ,Capacité des véhicules: 70,100,120,250,250 unités
- ✓ Demande pour chaque client: générée aléatoirement entre 1 et 13 unités
- ✓ Distance Calculer d'après les localisations

### 3. Méthodologie :

Pour résoudre le problème de VRP en utilisant Excel VBA et Solver avec le moteur COIN-OR CBC, nous suivrons les étapes suivantes :

#### Paramétrage :

- ✓ Max Solution Time : 30 s
- ✓ Tolérance : 0%
- ✓ Engine :
- ✓ COIN-OR
- ✓ CBC

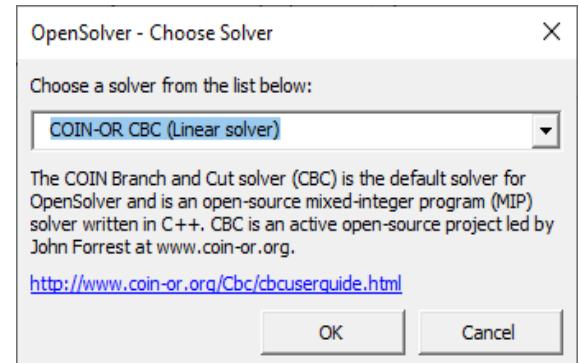


Figure 26 paramétrage de open solver

### Étape 1: Entrée de données

Tout d'abord, nous entrerons les données du problème dans Excel. Nous créerons un tableau avec les colonnes suivantes:

POINT NUMBER : Entre Stop Point Number

TRACK NUMBER : Entre Track Number

CAPACITY : Entre Tracks's Capacity

MAX QUANTITY : Entre maximum Quantity

**Submit**

Nbr des villes , Trucks Nbr (Contraint Désactivé) , Capacité , Quantité Max

**Étape 2 :** Générer toutes les matrices nécessaires : matrice de distances, matrice booléenne, matrice de contraintes et matrice de flux.

**Figure 27 Matrice des contraintes**

|                 | Quida-Ar | Nador   | Driouch  | Jerada   | Berkane  | Guerouf  | Figuig   | Marrakech | Chichaoua | Al Haouz  | Kelâa des | Essaouira | Rehamma  | Safi     | Yousseffia | Errachidia  | Duarzazate  |             |
|-----------------|----------|---------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|-----------|----------|----------|------------|-------------|-------------|-------------|
| Distance Matrix | 34.66833 | 35.207  | 34.97694 | 34.31053 | 34.92668 | 34.22558 | 32.70688 | 31.62593  | 31.54683  | 31.51293  | 32.05433  | 31.51183  | 34.92133 | 32.26508 | 32.245801  | 31.929069   | 30.920193   |             |
| Dourda-And      | 34.66833 | -2.0362 | 46.4964  | 10.96376 | 48.45634 | 25.68784 | 34.63287 | 56.16793  | 58.06749  | 40.80746  | 50.11174  | 16.30598  | 53.10475 | 35.62608 | 15.0131    | 40.5238097  | 42.9849088  | 48.48568017 |
| Nador           | 35.207   | -2.9125 | 45.49644 | 50.68688 | 84.197   | 48.90683 | 38.60552 | 102.1401  | 101.493   | 30.32474  | 94.56579  | 30.29367  | 58.32908 | 74.70217 | 56.83967   | 79.014071   | 12.4653479  | 85.5572913  |
| Driouch         | 34.97694 | -3.391  | 10.96376 | 50.68688 | 56.02074 | 36.67143 | 45.13249 | 51.95446  | 51.76636  | 31.49175  | 52.65839  | 21.52839  | 44.54944 | 42.64954 | 6.172745   | 47.4721384  | 44.5198949  | 55.5252697  |
| Jerada          | 34.3105  | -2.16   | 48.45634 | 84.197   | 56.02074 | 35.35688 | 49.1413  | 58.04313  | 67.93652  | 63.3052   | 46.18482  | 59.9822   | 100.5684 | 34.87175 | 55.00981   | 9.02487475  | 85.9721585  | 3.16292217  |
| Berkane         | 34.9267  | -2.3294 | 25.68784 | 48.90683 | 38.60552 | 36.67143 | 35.35688 | 15.76936  | 69.96852  | 75.42971  | 63.83312  | 50.80268  | 28.79854 | 76.58743 | 27.79647   | 39.67202    | 33.568542   | 51.9652539  |
| Guerouf         | 34.2256  | -3.3523 | 34.63287 | 38.60552 | 45.13249 | 49.1413  | 15.76936 | 84.89116  | 93.53011  | 75.24626  | 66.52365  | 29.50817  | 78.55611 | 43.24515 | 49.46162   | 42.4072543  | 44.9671747  | 51.1376505  |
| Figuig          | 32.7069  | -2.6288 | 56.16793 | 102.14   | 51.95446 | 58.04313 | 69.93652 | 84.89116  | 11.85489  | 32.35212  | 32.65902  | 72.02372  | 81.87956 | 49.95393 | 43.19627   | 51.1576207  | 96.4598509  | 85.9312019  |
| Marrakech       | 31.6256  | -7.9895 | 56.06749 | 102.419  | 57.16766 | 67.93652 | 75.42971 | 69.53011  | 11.85489  | 25.87473  | 43.75444  | 72.96168  | 75.36267 | 58.54689 | 45.95393   | 60.37451656 | 95.45463343 | 65.24652426 |
| Chichaoua       | 31.5469  | -8.7875 | 40.80478 | 80.3247  | 31.49175 | 69.3052  | 63.83919 | 75.24626  | 32.35212  | 25.87473  | 52.60889  | 72.95881  | 49.78860 | 56.81254 | 25.95125   | 60.423236   | 71.9786312  | 67.4417382  |
| Al Haouz        | 31.0251  | -9.7678 | 50.11184 | 94.56858 | 52.65839 | 26.94182 | 50.80268 | 65.62956  | 32.69502  | 43.75944  | 52.60889  | 65.68534  | 94.40005 | 23.86202 | 48.82686   | 22.4795388  | 92.8568159  | 33.28734874 |
| Kellâa de       | 32.0543  | -7.4068 | 16.30598 | 30.2397  | 21.52369 | 59.97822 | 28.79854 | 29.05817  | 72.02372  | 72.95881  | 57.93293  | 65.68534  | 49.25208 | 48.42996 | 27.5885    | 53.193274   | 27.2081018  | 60.3263976  |
| Essaouira       | 31.5185  | -9.7621 | 53.10475 | 58.3291  | 45.54594 | 100.5684 | 76.58743 | 78.56611  | 81.87956  | 75.36267  | 49.78860  | 94.40005  | 49.25208 | 87.174   | 46.35656   | 92.0011631  | 45.9637485  | 100.0523402 |
| Rehamma         | 34.9314  | -2.1438 | 35.62608 | 74.7022  | 42.64954 | 13.48719 | 27.79474 | 43.24515  | 49.95393  | 55.56851  | 66.21254  | 23.86202  | 48.42996 | 87.174   | 41.52191   | 4.09750812  | 75.1717886  | 12.9125475  |
| Safi            | 32.2651  | -9.2306 | 15.013   | 56.8337  | 6.172745 | 55.00891 | 39.67202 | 49.64162  | 45.59393  | 25.95125  | 48.82686  | 27.5885   | 46.35656 | 41.52191 | 46.1759314 | 50.1592403  | 54.17325042 | 73.80456885 |
| Yousseffia      | 32.2484  | -8.5324 | 40.52381 | 73.0141  | 47.47217 | 9.028475 | 31.35848 | 46.42057  | 51.15767  | 60.37432  | 60.41223  | 22.47495  | 53.18323 | 92.00116 | 4.9075681  | 46.175933   | 8.005361822 | 86.94405495 |
| Errachidia      | 31.9291  | -4.4341 | 42.96431 | 12.4669  | 44.51986 | 85.97216 | 51.95624 | 44.96671  | 96.45968  | 95.454658 | 71.79863  | 82.56881  | 27.20815 | 45.96375 | 75.1779    | 50.5182     | 79.80456885 | 86.94405495 |
| Duarzazate      | 30.9202  | -6.5109 | 48.45608 | 85.51657 | 55.52527 | 3.96221  | 37.00281 | 51.19763  | 55.1932   | 65.24652  | 67.44738  | 23.93373  | 60.6522  | 100.0525 | 12.91254   | 54.77325    | 8.053161822 | 86.94405495 |

**Figure 28 Matrice des distances**

**Figure 29 Matrix Boolean**

| Fj       | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 | Point 10 | Point 11 | Point 12 | Point 13 | Point 14 | Point 15 | Point 16 | Point 17 | Flow In | Flow Out | Balance |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|----------|----------|----------|----------|----------|----------|----------|---------|----------|---------|
| Point 1  | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0       | 122      | 0       |
| Point 2  | 0       | 0       | 0       | 0       | 0       | 112     | 0       | 0       | 0       | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 112     | 103      | 0       |
| Point 3  | 0       | 103     | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 103     | 98       | 0       |
| Point 4  | 0       | 0       | 0       | 0       | 0       | 0       | 8       | 0       | 0       | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 8        | 0       | 0        | 0       |
| Point 5  | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0        | 0        | 0        | 115      | 0        | 0        | 0        | 0        | 115     | 115      | 0       |
| Point 6  | 0       | 0       | 88      | 0       | 0       | 0       | 0       | 0       | 0       | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 88      | 94       | 0       |
| Point 7  | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0        | 0        | 0        | 0        | 0        | 0        | 16       | 0        | 16      | 8        | 0       |
| Point 8  | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 43      | 0       | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 43      | 31       | 0       |
| Point 9  | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0        | 0        | 51       | 0        | 0        | 0        | 0        | 0        | 51      | 43       | 0       |
| Point 10 | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0        | 0        | 0        | 59       | 0        | 0        | 0        | 0        | 59      | 55       | 0       |
| Point 11 | 0       | 0       | 0       | 0       | 0       | 94      | 0       | 0       | 0       | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 94      | 84       | 0       |
| Point 12 | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 55       | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 55      | 51       | 0       |
| Point 13 | 122     | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 122     | 115      | 0       |
| Point 14 | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 70      | 59       | 0       |
| Point 15 | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 84      | 70       | 0       |
| Point 16 | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 25      | 16       | 0       |
| Point 17 | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 31      | 25       | 0       |
|          | 122     | 103     | 88      | 0       | 112     | 94      | 8       | 31      | 43      | 55       | 84       | 51       | 115      | 59       | 70       | 16       | 25       |         |          |         |

Figure 30 Matrix Flow

### Étape 3 : Entrer le modèle dans Open Solver.

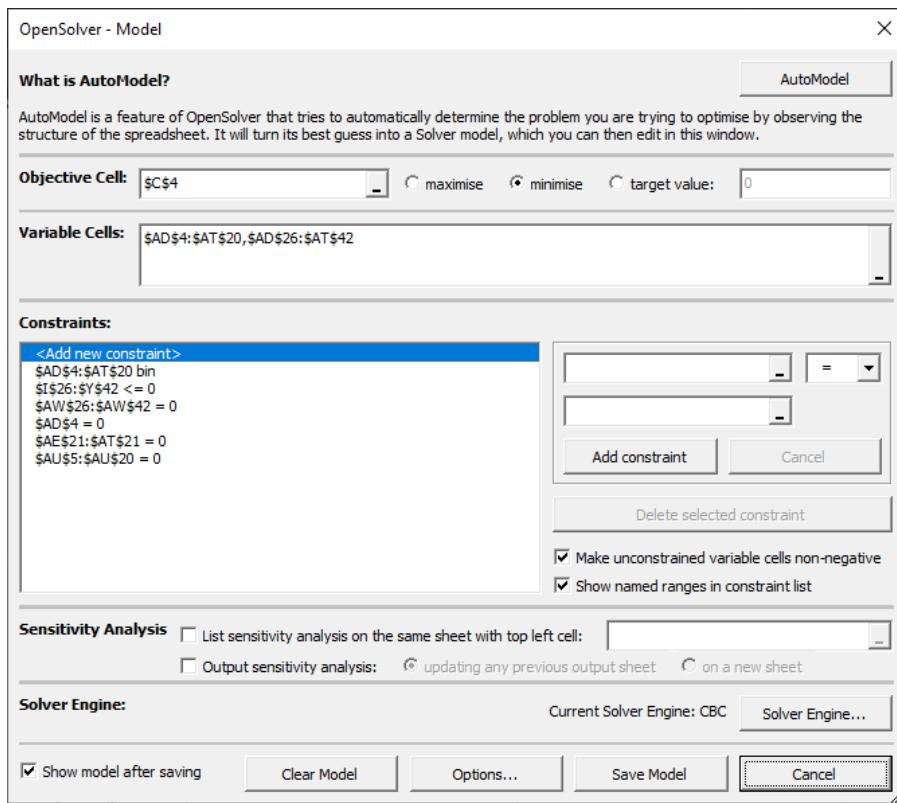
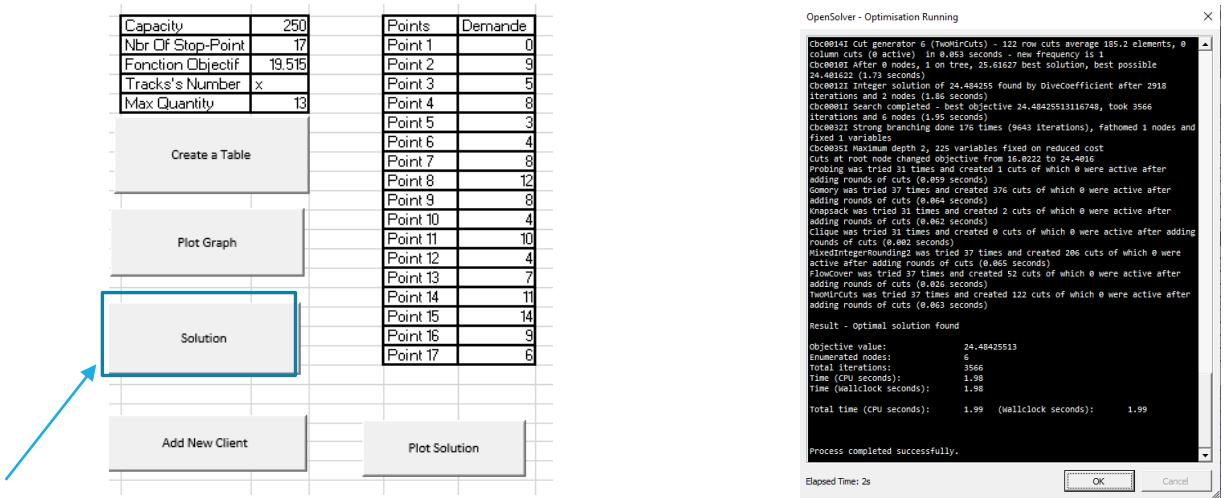


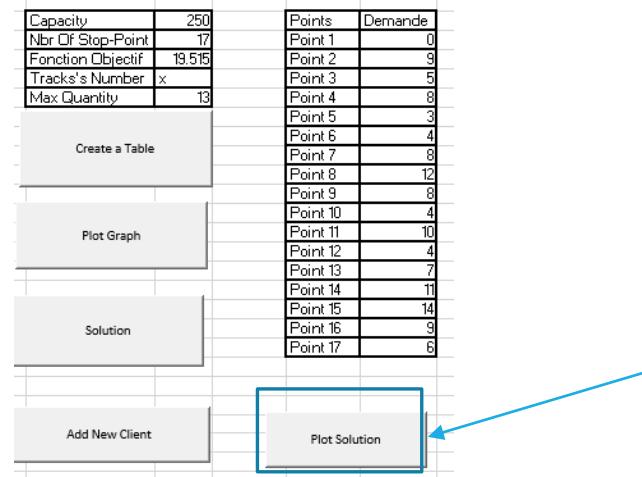
Figure 31 : Entrer le modèle dans Open Solver.

### Étape 4 : Lancer la résolution du problème en utilisant le moteur COIN OR CBC.



**Étape 5 :** Cliquer sur le bouton "Plot Solution" pour afficher la solution obtenue.

Pour présenter les résultats de manière claire, vous pouvez utiliser des graphiques et des tableaux pour représenter les itinéraires, les distances parcourues, les demandes satisfaites, etc.



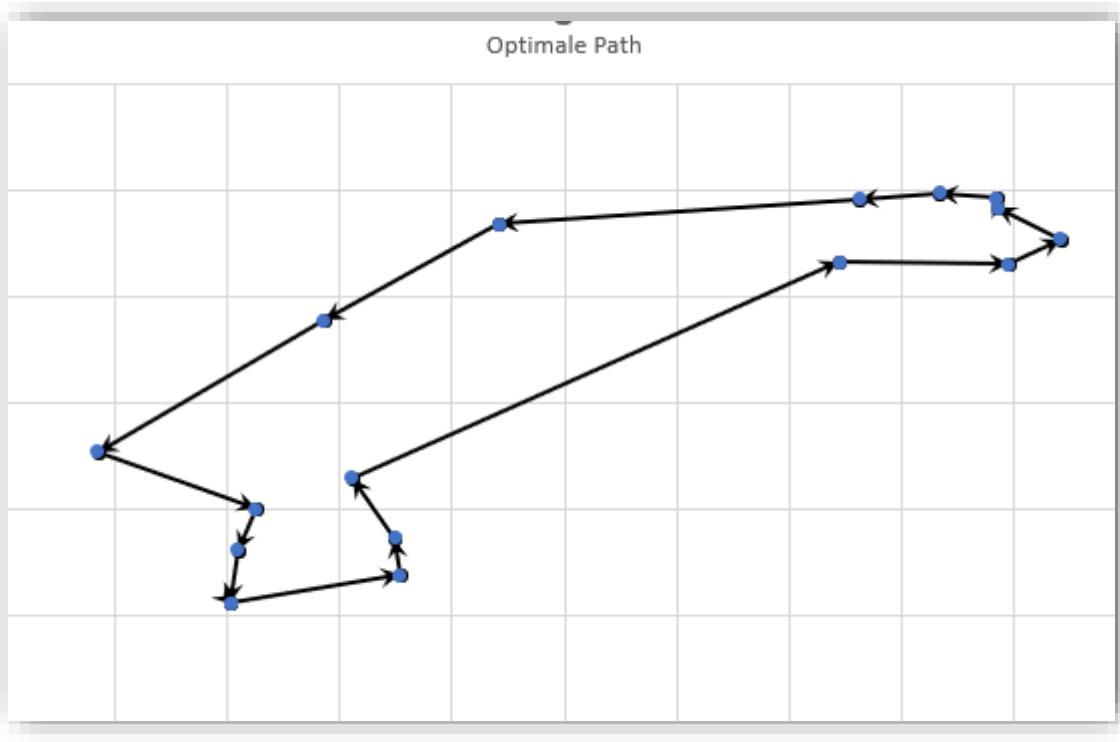


Figure 32 représentation graphique de la Solution

#### 4. Résolution :

La solution précédemment présentée ne peut pas traiter des problèmes de grande taille. Afin de pallier ce problème, il a été décidé de diviser les clients en cinq régions et d'appliquer les étapes précédentes pour chaque région. Les résultats obtenus sont les suivants

## Région 1 :

|                   |               |
|-------------------|---------------|
| Capacity          | 250           |
| Nbr Of Stop-Point | 17            |
| Fonction Objectif | 19.515        |
| Tracks's Number   | x             |
| Max Quantity      | 13            |
| Create a Table    |               |
| Plot Graph        |               |
| Solution          |               |
| Add New Client    | Plot Solution |

### OpenSolver - Optimisation Running

```
Cbc0014I Cut generator 6 (TwoMircuts) - 122 row cuts average 185.2 elements, 0 column cuts (0 active) in 0.053 seconds - new frequency is 1
Cbc0010I After 0 nodes, 1 on tree, 25.61627 best solution, best possible 24.401622 (1.73 seconds)
Cbc0012I Integer solution of 24.484255 found by DiveCoefficient after 2918 iterations and 2 nodes (1.86 seconds)
Cbc0001I Search completed - best objective 24.48425513116748, took 3566 iterations and 6 nodes (1.98 seconds)
Cbc0032I Strong branching done 176 times (9643 iterations), fathomed 1 nodes and fixed 1 variables
Cbc0035I Maximum depth 2, 225 variables fixed on reduced cost
Cuts at root node changed objective from 16.0222 to 24.4016
Probing was tried 31 times and created 1 cuts of which 0 were active after adding rounds of cuts (0.059 seconds)
Gomory was tried 37 times and created 376 cuts of which 0 were active after adding rounds of cuts (0.064 seconds)
Knapsack was tried 31 times and created 2 cuts of which 0 were active after adding rounds of cuts (0.062 seconds)
Cliques was tried 31 times and created 0 cuts of which 0 were active after adding rounds of cuts (0.002 seconds)
MixedIntegerRounding2 was tried 37 times and created 206 cuts of which 0 were active after adding rounds of cuts (0.065 seconds)
FlowCover was tried 37 times and created 52 cuts of which 0 were active after adding rounds of cuts (0.026 seconds)
TwoMircuts was tried 37 times and created 122 cuts of which 0 were active after adding rounds of cuts (0.063 seconds)

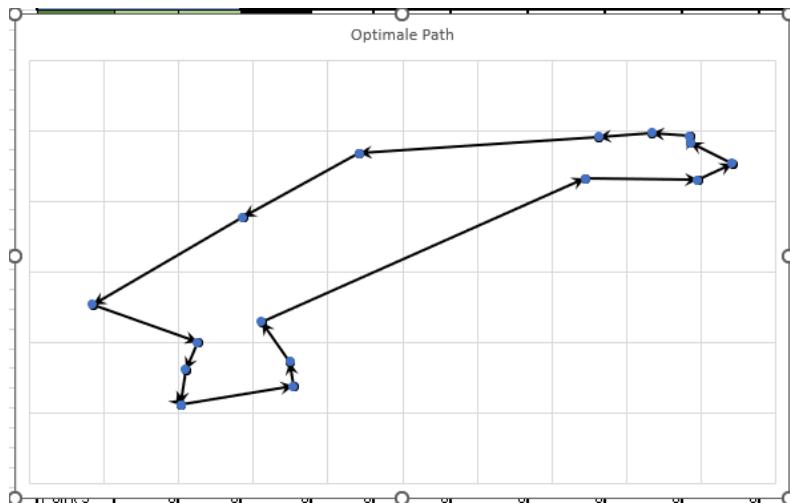
Result - optimal solution found

Objective value: 24.48425513
Enumerated nodes: 6
Total iterations: 3566
Time (CPU seconds): 1.98
Time (Wallclock seconds): 1.98

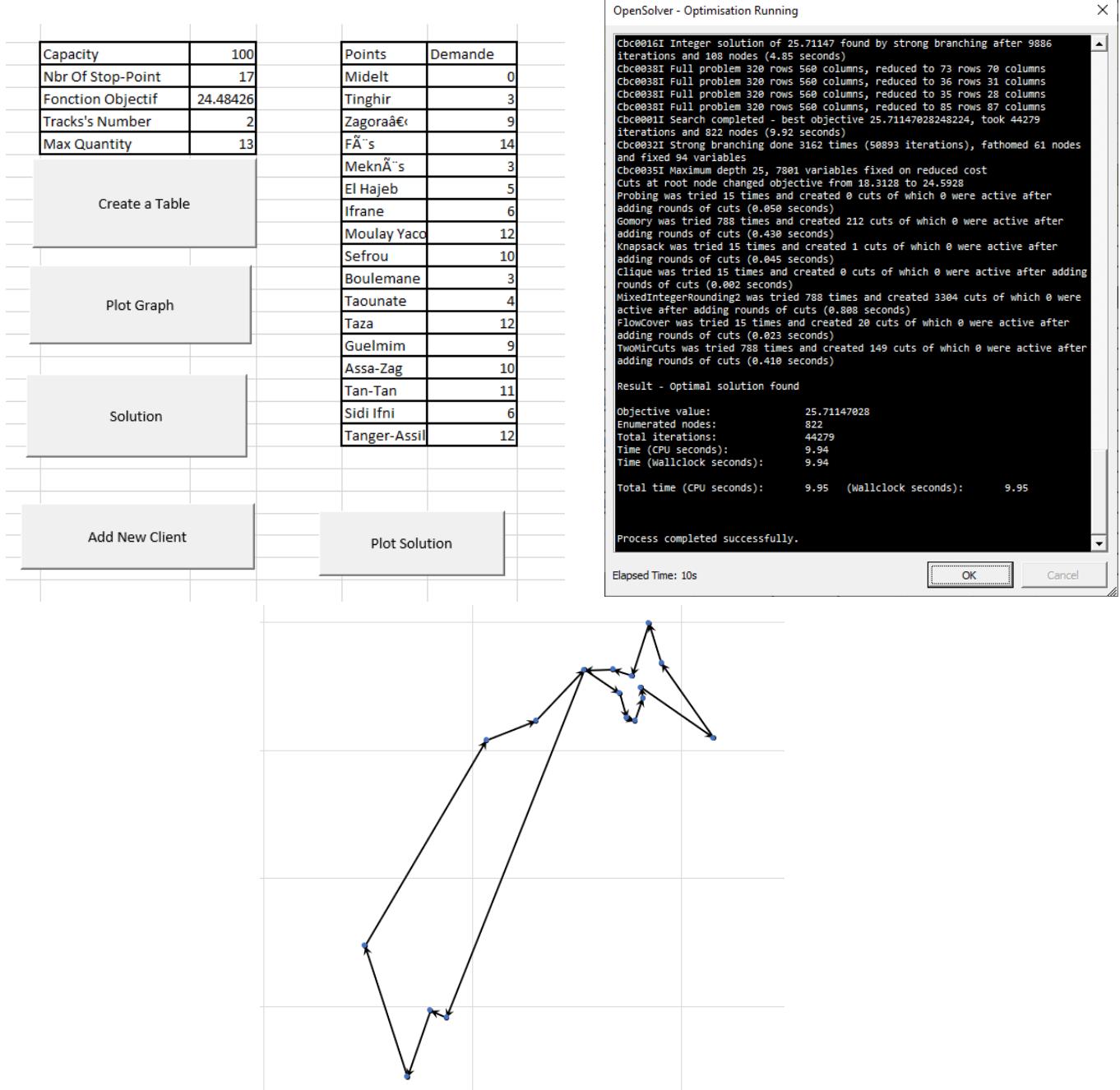
Total time (CPU seconds): 1.99 (Wallclock seconds): 1.99

Process completed successfully.
```

Elapsed Time: 2s



## Région 2 :



### Région 3:

|                   |        |
|-------------------|--------|
| Capacity          | 120    |
| Nbr Of Stop-Point | 17     |
| Fonction Objectif | 25.711 |
| Tracks's Number   | 2      |
| Max Quantity      | 13     |

Create a Table

| Points      | Demande |
|-------------|---------|
| Midelt      | 0       |
| Tinghir     | 3       |
| Zagoraâ€    | 9       |
| FA's        | 14      |
| MeknÃ's     | 3       |
| El Hajeb    | 5       |
| Ifra        | 6       |
| Moulay Yac  | 12      |
| Sefrou      | 10      |
| Boulemane   | 3       |
| Taounate    | 4       |
| Taza        | 12      |
| Guelmim     | 9       |
| Assa-Zag    | 10      |
| Tan-Tan     | 11      |
| Sidi Ifni   | 6       |
| Tanger-Assi | 12      |

Plot Graph

Solution

Add New Client

Plot Solution

OpenSolver - Optimisation Running

```
Cbc00381 Full problem 320 rows 560 columns, reduced to 92 rows 96 columns
Cbc00381 Full problem 320 rows 560 columns, reduced to 70 rows 67 columns
Cbc00381 Full problem 320 rows 560 columns, reduced to 69 rows 65 columns
Cbc00381 Full problem 320 rows 560 columns, reduced to 48 rows 47 columns
Cbc00101 After 1000 nodes, 1 on tree, 20.02354 best solution, best possible
19.989426 (14.06 seconds)
Cbc00011 Search completed - best objective 20.0235426559179, took 49899
iterations and 1012 nodes (14.14 seconds)
Cbc00321 Strong branching done 3496 times (54124 iterations), fathomed 86 nodes
and fixed 206 variables
Cbc00351 Maximum depth 25, 6655 variables fixed on reduced cost
Cuts at root node changed objective from 15.1944 to 18.8452
Probing was tried 47 times and created 1 cuts of which 0 were active after
adding rounds of cuts (0.109 seconds)
Gomory was tried 1196 times and created 944 cuts of which 0 were active after
adding rounds of cuts (0.806 seconds)
Knapsack was tried 47 times and created 1 cuts of which 0 were active after
adding rounds of cuts (0.124 seconds)
Cliques was tried 47 times and created 0 cuts of which 0 were active after adding
rounds of cuts (0.007 seconds)
MixedIntegerRounding2 was tried 1196 times and created 5130 cuts of which 0 were
active after adding rounds of cuts (1.414 seconds)
FlowCover was tried 47 times and created 43 cuts of which 0 were active after
adding rounds of cuts (0.084 seconds)
TwoMircuts was tried 1196 times and created 156 cuts of which 0 were active
after adding rounds of cuts (0.540 seconds)

Result - Optimal solution found

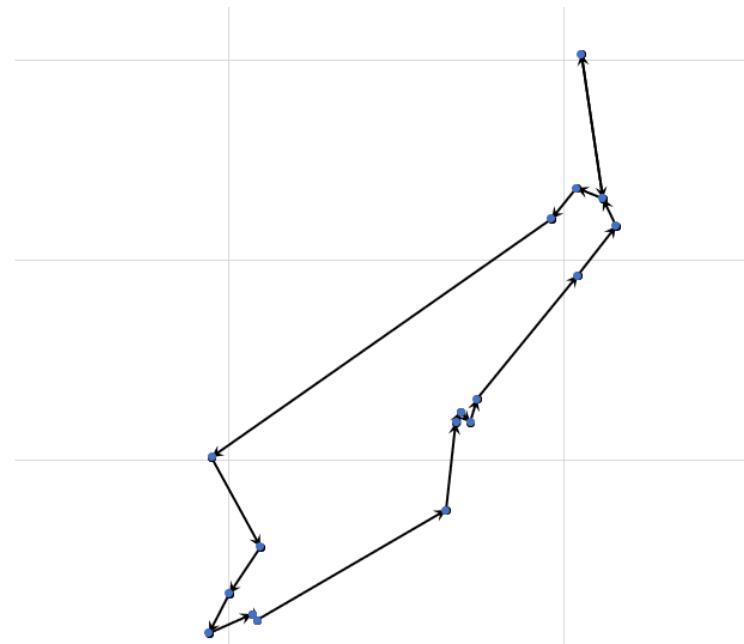
Objective value: 20.02354266
Enumerated nodes: 1012
Total iterations: 49899
Time (CPU seconds): 14.20
Time (Wallclock seconds): 14.20

Total time (CPU seconds): 14.21 (Wallclock seconds): 14.21

Process completed successfully.
```

Elapsed Time: 14s

OK Cancel



## Région 4 :

|                   |          |
|-------------------|----------|
| Capacity          | 250      |
| Nbr Of Stop-Point | 17       |
| Fonction Objectif | 33.08841 |
| Tracks's Number   | 2        |
| Max Quantity      | 13       |

Create a Table

| Points       | Demande |
|--------------|---------|
| Benslimane   | 0       |
| Berrechid    | 11      |
| Settat       | 8       |
| Sidi Bennour | 10      |
| Oued Ed-Dah  | 5       |
| Aousserdâ€   | 9       |
| BÃ©ni Mellal | 11      |
| Azilal       | 8       |
| Equih Ben Sa | 14      |
| KhÃ©nifra    | 13      |
| Khouribgaâ€  | 5       |
| Rabat        | 12      |
| SalÃ©        | 12      |
| Skhirate-TÃ© | 4       |
| KÃ©nitra     | 3       |
| KhÃ©misset   | 11      |
| Sidi Kacem   | 12      |

Plot Graph

Solution

(Ctrl)

OpenSolver - Optimisation Running

```
Cbc0014I cut generator 5 (FlowCover) - 31 row cuts average 18.3 elements, 0 column cuts (0 active) in 0.011 seconds - new frequency is -100
Cbc0014I cut generator 6 (TwoMircuts) - 178 row cuts average 124.1 elements, 0 column cuts (0 active) in 0.080 seconds - new frequency is 1
Cbc0010T After 0 nodes, 1 on tree, 33.088406 best solution, best possible 32.948714 (1.70 seconds)
Cbc0001T Search completed - best objective 33.08840618656118, took 1610 iterations and 2 nodes (1.78 seconds)
Cbc0032T Strong branching done 112 times (3992 iterations), fathomed 1 nodes and fixed 0 variables
Cbc0035T Maximum depth 0, 232 variables fixed on reduced cost
Cuts at root node changed objective from 11.9872 to 32.9487
Probing was tried 26 times and created 59 cuts of which 0 were active after adding rounds of cuts (0.038 seconds)
Gomory was tried 26 times and created 441 cuts of which 0 were active after adding rounds of cuts (0.034 seconds)
Knapsack was tried 21 times and created 2 cuts of which 0 were active after adding rounds of cuts (0.051 seconds)
Clasp was tried 21 times and created 0 cuts of which 0 were active after adding rounds of cuts (0.203 seconds)
MixedIntegerRounding2 was tried 26 times and created 169 cuts of which 0 were active after adding rounds of cuts (0.016 seconds)
FlowCover was tried 21 times and created 31 cuts of which 0 were active after adding rounds of cuts (0.011 seconds)
TwoMircuts was tried 26 times and created 217 cuts of which 0 were active after adding rounds of cuts (0.098 seconds)

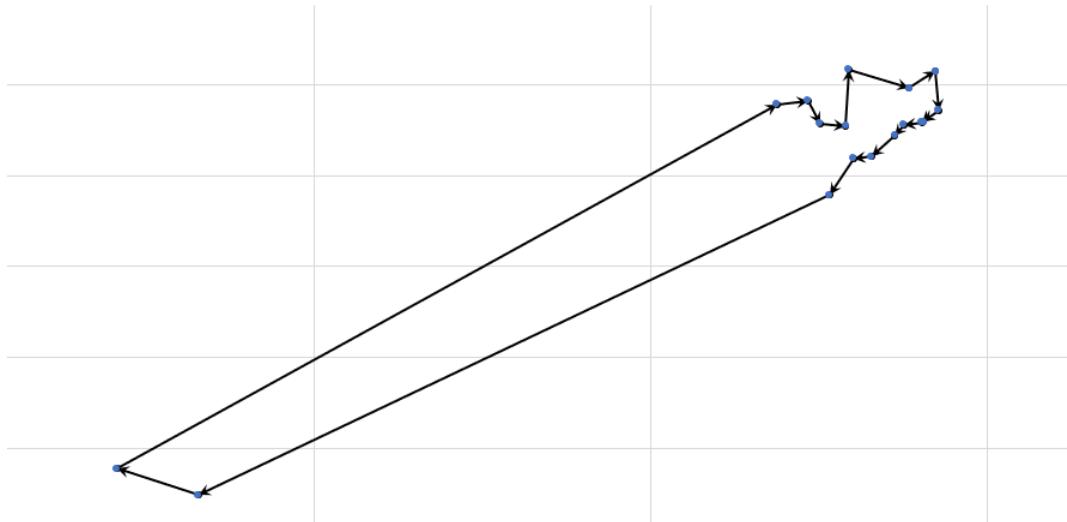
Result - Optimal solution found

Objective value: 33.08840619
Enumerated nodes: 2
Total iterations: 1610
Time (CPU seconds): 1.80
Time (Wallclock seconds): 1.80

Total time (CPU seconds): 1.81 (Wallclock seconds): 1.81
```

Process completed successfully.

Elapsed Time: 2s



## Région 5 :

|                   |         |
|-------------------|---------|
| Capacity          | 250     |
| Nbr Of Stop-Point | 5       |
| Fonction Objectif | 24.2499 |
| Tracks's Number   | 3       |
| Max Quantity      | 13      |
| Create a Table    |         |

| Points       | Demande |
|--------------|---------|
| Sidi Slimane | 0       |
| Laâyoune     | 9       |
| Boujdour     | 8       |
| Tarfaya      | 7       |
| Es-Semaraâ   | 5       |

OpenSolver - Optimisation Running

```
Cbc0014I Cut generator 3 (clique) - 0 row cuts average 0.0 elements, 0 column cuts (0 active) in 0.000 seconds - new frequency is -100
Cbc0014I Cut generator 4 (MixedIntegerRounding2) - 7 row cuts average 9.1 elements, 0 column cuts (0 active) in 0.000 seconds - new frequency is 1
Cbc0014I Cut generator 5 (FlowCover) - 1 row cuts average 6.0 elements, 0 column cuts (0 active) in 0.000 seconds - new frequency is -100
Cbc0014I Cut generator 6 (TwoMircuts) - 25 row cuts average 9.6 elements, 0 column cuts (0 active) in 0.000 seconds - new frequency is -100
Cbc00011 Search completed - best objective 24.249899348548725, took 29 iterations and 0 nodes (0.09 seconds)
Cbc0035I Maximum depth 0, 6 variables fixed on reduced cost
Cuts at root node changed objective from 8.74434 to 24.2499
Probing was tried 3 times and created 17 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
Gomory was tried 3 times and created 10 cuts of which 0 were active after adding rounds of cuts (0.001 seconds)
Knapsack was tried 3 times and created 0 cuts of which 0 were active after adding rounds of cuts (0.002 seconds)
Cliques was tried 3 times and created 0 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
MixedIntegerRounding2 was tried 3 times and created 7 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
FlowCover was tried 3 times and created 1 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
TwoMircuts was tried 3 times and created 25 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)

Result - Optimal solution found

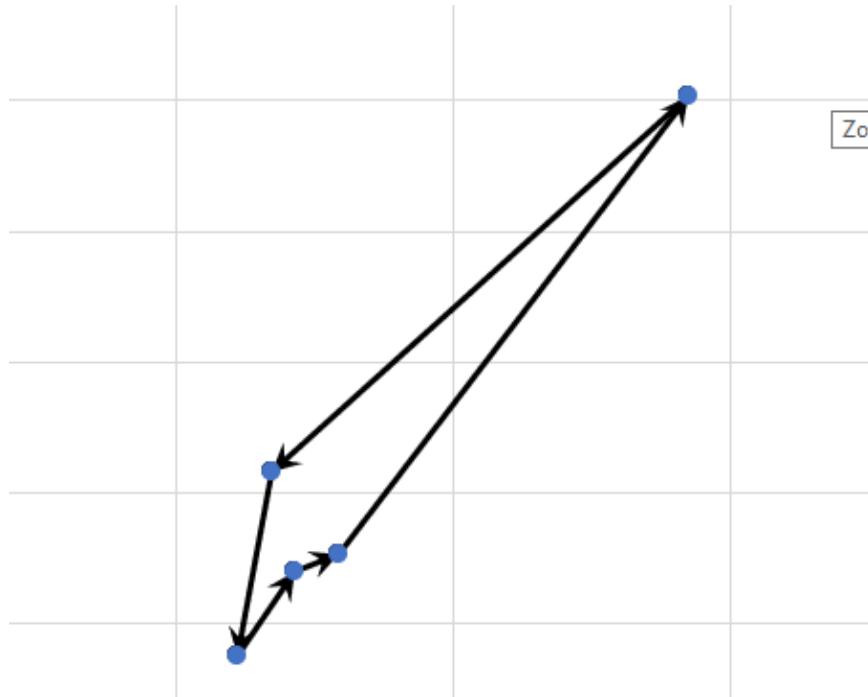
Objective value: 24.249899348548725
Enumerated nodes: 0
Total iterations: 29
Time (CPU seconds): 0.14
Time (Wallclock seconds): 0.14

Total time (CPU seconds): 0.14 (Wallclock seconds): 0.14

Process completed successfully.
```

Elapsed Time: 0s

OK Cancel



## 5. Conclusion:

Dans l'ensemble, les trois solutions présentées dans cette partie ont leurs propres avantages et inconvénients en fonction des besoins spécifiques du problème VRP. La méthode de résolution exacte offerte par le code Python basé sur la bibliothèque Pulp peut garantir des résultats optimaux pour des problèmes VRP de petite et moyenne taille. Les méthodes heuristiques et métaheuristiques offertes par OpenDoor Logistic Studio peuvent être plus appropriées pour des problèmes VRP de grande taille, où une solution exacte peut être trop coûteuse en termes de temps de calcul. La solution basée sur VBA d'Excel peut être utile pour des utilisateurs familiers avec Excel qui souhaitent résoudre des problèmes VRP de petite et moyenne taille de manière simple et rapide. En fin de compte, le choix de la méthode de résolution dépendra des exigences spécifiques du problème VRP à résoudre, ainsi que des compétences et des ressources disponibles pour l'utilisateur.

## Conclusion générale :

Après avoir mené à bien un projet d'optimisation de VRP (Vehicle Routing Problem), plus précisément optimisation de trajet d'un ensemble de véhicules passant par 96 province au Maroc, L'objectif principal de cette optimisation est de trouver une solution efficace et rentable pour le transport de marchandises ou de personnes, en minimisant les coûts de transport et en respectant les contraintes de temps et de capacité.

L'algorithme Branch and Bond utilisé comme approche exacte nous a permis de garantir l'optimalité de la solution, mais un petit peu coûteux en termes de temps de calcul pour les problèmes de grande taille. L'algorithme du Plus Proche Voisin offre un temps rapide mais n'atteint pas l'optimal et L'algorithme Génétique nécessite plus de temps de calcul et des résultats pour une grande population. Donc La sélection de l'algorithme approprié doit être basée sur une évaluation approfondie des exigences du problème et des contraintes du contexte d'application.

En fin, l'optimisation de VRP a permis d'améliorer l'efficacité et la rentabilité du transport de marchandises ou de personnes, Cependant, il est important de continuer à surveiller et à améliorer les méthodes d'optimisation utilisées pour maintenir une efficacité et une rentabilité maximales à long terme.