

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

Centre Universitaire  
Abd Elhafid Boussouf Mila

Institut des Sciences et Technologie

Département de Mathématiques et Informatique

## Mémoire préparé en vue de l'obtention du diplôme de Master

En : Informatique

Spécialité : Sciences et Technologies de l'Information et de la Communication  
(STIC)

**Un outil informatique d'ordonnancement et de  
planification des interventions des équipes de la  
société de SONELGAZ**

Préparé par :

Bensaci Ahmed Yacine

Fedali Imad

Devant le jury

Mr. Merabet Adil	MAA	C.U.Abd Elhafid Boussouf	Président
Mr. Kimouche Abdelkader	MAA	C.U.Abd Elhafid Boussouf	Rapporteur
Mr. Boukhchem Nadir	MAA	C.U.Abd Elhafid Boussouf	Examineur

Année Universitaire : 2016/2017

# REMERCIEMENT

*Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce Modeste travail.*

*Nous présentons nos sincères gratitudes et remerciements à notre encadreur Kimouche Abdelkader pour le grand soutien morale qu'il nous a apporté au cours de notre projet et aussi pour son aide précieuse et ses conseils judicieux qu'il nous a fait bénéficier.*

*Nos remerciements très sincères, vont aussi à tous les membres du personnel du département informatique, et en particulier, à tous ceux qui nous ont enseigné durant toutes nos années d'études universitaires. Qu'il trouve ici l'assurance de notre profonde gratitude.*

*En fin, à tous ceux qui ont contribué de près ou de loin à l'achèvement de ce travail.*



**DÉDICACE**

**C'est avec joie que je dédie ce**

**modeste travail :**

**À mes très chers parents qui**

**j'espère rendre fière, pour**

**leurs patiences et**

**encouragements. Que dieu les**

**protègent.**

**À ma femme et l'amour de ma**

**vie Nour.**

**À mes amis et mes collègues**

**pour les bons moments que j'ai**

**passé avec eux**

Yacine



# DÉDICACE

*Je tiens en tout premier lieu à  
remercier le dieu, je voudrais dédie ce  
modeste travail*

*A mes très chers parents  
Pour tout l'amour dont vous m'avez  
entouré, pour tout ce que vous avez fait  
pour moi.*

*Je ferai de mon mieux pour rester un  
sujet de fierté à vos yeux avec l'espoir  
de ne jamais vous décevoir*

*A mes très chers frères : Hicham, Adel  
et mes sœurs Rima, Bouthaina, Amani  
Vous occupez une place particulière dans  
mon cœur. Je vous dédie ce travail en  
vous souhaitant un avenir radieux, plein  
de bonheur et de succès*

*A mes chères amies :  
Seif, Ali, Younes, Bouaalam, Mohamed,  
Amar, Iskandar, Djad;*

*Désolé si j'oublie  
En souvenir de nos éclats de rire, des  
bons moments et nuits blanches. En  
souvenir de tout ce qu'on a vécu  
ensemble. J'espère de tout mon cœur  
Que notre amitié durera éternellement*

**Imad**



## **Introduction Générale**

### **chapitre1 : La résolution des problèmes d'optimisation combinatoire**

1.1 Introduction .....	1
1.2 Problème d'optimisation combinatoire.....	1
1.2.1 Définition.....	1
1.3 Exemples.....	2
1.3.1 Le problème du voyageur de commerce.....	2
1.3.2 Le problème de Sac-à-dos.....	3
1.4 Résolution d'un problème d'optimisation combinatoire.....	3
1.5 Les méthodes d'optimisation combinatoire.....	4
1.5.1 Les méthodes exactes.....	4
1.5.1.1 La programmation dynamique.....	5
1.5.1.2 La méthode de simplexe.....	5
1.5.1.3 Branch and Bound.....	5
1.5.2 Les méthodes approchées.....	6
1.5.2.1 Heuristique.....	6
1.5.2.2 Méta-Heuristique.....	6
1.5.2.2.1 La Recuit simulé.....	7
1.5.2.2.2 La recherche Tabou.....	9
1.5.2.2.3 Colonies de fourmis.....	9
1.5.2.2.4 La méthode GRASP.....	9
1.5.2.2.5 L'algorithme génétique.....	9
1.6 Conclusion.....	10

### **Chapitre 2 : L'algorithme génétique**

2.1 Introduction.....	12
2.2 Définition .....	12
2.3 Fonctionnement de l'algorithme.....	12
2.4 La structure.....	15
2.5 Codage des individus.....	15
2.6 L'opérateur de sélection .....	16
a) Méthode loterie biaisée.....	16
b) Méthode Elitiste.....	17

c) Méthode Tournoi.....	17
d) Méthode Stochastique.....	18
2.7 L'opérateur de croisement.....	18
a. Croisement à un point .....	18
b. Croisement multipoints.....	18
c. Croisement uniforme.....	19
2.8 L'opérateur de mutation.....	19
2.9 L'opérateur de remplacement.....	21
a) Stationnaire.....	21
b) Elitiste.....	21
2.10 Conclusion.....	22

### **Chapitre 3 : Étude de cas : problème d'interventions des équipes de**

#### **Sonelgaz**

3.1 Introduction.....	24
3.2 Présentation de Sonelgaz .....	24
3.3 DD Mila.....	24
3.4 Présentation du problème.....	25
3.5 TAD.....	26
3.6 Algorithme génétique développé .....	27
a. Codage de l'individu.....	29
b. Population initiale.....	29
c. Mésure de qualité.....	30
d. La sélection.....	30
e. Le Croisement.....	31
f. La mutation .....	32
g. La condition d'arrêt.....	32
3.7 Conclusion.....	33

## **Chapitre 4 : Implémentation et analyse expérimentale**

4.1 Introduction.....	35
4.2 Plateforme logiciel.....	35
4.3 Langage java.....	35
4.4 SGBDMySQL .....	36
4.5 Description des interfaces de l'application .....	38
4.6 Accueil de l'application .....	38
4.7 Testes et comparaison.....	41
4.8 Analyse des résultats.....	47
4.9 Conclusion.....	47
<b>Conclusion générale.....</b>	<b>48</b>

## Table de figures :

Figure 1.1 : L'optimisation combinatoire.....	1
Figure 2 : Le problème de voyageur de commerce. ....	2
Figure 3 Le problème de Sac-à-dos .....	3
Figure 4 Classification des méthodes d'optimisation combinatoire.....	4
Figure 5 Divisé en sous-problèmes.....	5
Figure 6 Fonctionnement de l'algorithme de recuit simulé.....	8
Figure 7 La procédure générale de l'algorithme génétique .....	12
Figure 8 Codage des solutions.....	14
Figure 9 Sélection par roulette.....	15
Figure 10 Sélection par tournoi. ....	16
Figure 11 Croisement à un point. ....	17
Figure 12 Croisement à deux points. ....	18
Figure 13 Croisement uniforme.....	18
Figure 14 Une mutation.....	19
Figure 15 Exemple de problème à la demande.....	26
Figure 16 Exemple de codage.....	28
Figure4. 17 Table Ville de la base de données .....	35
Figure4. 18 Liste des villes de la Wilaya de Mila .....	36
Figure 4.19 Table Distance de la base de données .....	36
Figure 4.20 la fenêtre d'accueil de l'application .....	37
Figure 4.21 Ajout de demandes .....	38
Figure 4.22 Formulaire de saisie de paramètres de l'algorithme génétique.....	38
Figure 4.23 Carte de la ville .....	39
Figure4.24: remplir des informations .....	39
Figure 4.25 : exemple d'une solution du problème. ....	40
Figure 4.26 : requêtes de citoyens au service technique du SONELGAZ (Données réels d'une semaine) 41	41
Figure 4.27 Images de temps de travail .....	42
Figure 4.28 Les tournées appliquées par le groupe technique de la société.....	43
Figure 4.29 : L'ajout des demandes .....	44
Figure 4.30 Saisir les informations .....	44
Figure 4.31 les résultats de notre application. ....	45



## Introduction générale

« **L**e désir humain de perfection trouve son expression dans la théorie de l'optimisation. Elle étudie comment décrire et atteindre ce qui est meilleur, une fois que l'on connaît comment mesurer et modifier ce qui est bon et ce qui est mauvais . . . la théorie de l'optimisation comprend l'étude quantitative des optimums et les méthodes pour les trouver. » [2]

Il arrive très souvent que les sciences et les techniques imitent les mécanismes de la nature. Par exemple, le sonar ou le radar sont inspirés des techniques d'écholocation des dauphins ou des chauves-souris, la texture de la peau de requin a été imitée pour fabriquer des matériaux hydrodynamiques...

Les algorithmes génétiques (AG) proposent de reproduire les mécanismes d'évolution et d'adaptation génétique de la vie pour optimiser des problèmes complexes dont les données peuvent varier au cours du temps. La vie a besoin en permanence de s'adapter : à un nouvel environnement, à un nouveau prédateur, ou à de nouvelles proies. Le vocabulaire et les intuitions des algorithmes génétiques appartiennent en partie au vocabulaire et aux intuitions de la biologie. Ces algorithmes fabriquent des chromosomes qui codent chacun une solution potentielle à un problème donné. A chaque étape (appelée génération), ces chromosomes se combinent, mutent, et sont sélectionnés en fonction de leur qualité à répondre au problème. De même, dans la nature, selon la thèse darwinienne, seule la succession de croisements et de mutations aléatoires suffisent à expliquer l'adaptation des êtres vivants à leur milieu naturel (problème qui serait trop complexe à comprendre dans son ensemble). Dans les AG, la succession des croisements et des mutations permet d'arriver à une solution, qui sans être nécessairement optimale, peut être très satisfaisante. [3]

Dans le cadre de ce travail, on va s'intéresser à l'étude d'un problème très fréquent dans la vie quotidien des citoyens dans notre pays, c'est le problème de la baisse qualité de services offerts par les sociétés économique au niveau du territoire national. On va prendre comme cas d'étude les interventions des équipes de la société de SONALGAZ suite aux demandes des citoyens. Le temps de réponse moyen au demandes prend des jours voir même des semaines. Pour cela et dans le but d'aider les décideurs de la société à mieux exploiter les ressources humaines et matérielles, on a développé un outil informatique qui permet de répartir les demandes des citoyens sur les groupes techniques de la société d'une façon optimale,

l'application développée se base sur la méthode des algorithmes génétique, c'est une méta-heuristique qui permet de donner une solution efficace dans un temps raisonnable. [2]

Le reste de ce mémoire est organisé comme suit, dans le premier chapitre on va traiter les problèmes d'optimisation combinatoires, et leurs méthodes de résolutions existantes, dans le deuxième chapitre on va concentrer sur la méthode des algorithmes génétiques, ses étapes, et les techniques utilisées dans la littérature pour chaque étape. Le troisième chapitre décrit le problème des interventions des groupes technique de la société de SONALGAZ, et la méthode développée pour résoudre ce problème. Le quatrième chapitre présente les résultats expérimentaux des tests qu'on a fait sur des données réelles qu'on a accumulé à partir des agences de la société au niveau de la wilaya de Mila, une comparaison de notre solution avec la réalité appliquée montre l'importance de l'utilisation des techniques mathématique dans la résolution de ce genre de problème, et ouvre une porte sur l'étude des problèmes similaire.

# Chapitre 1 :

## **La résolution des problèmes d'optimisation combinatoires**

## 1.1 Introduction

L'optimisation combinatoire est un outil indispensable combinant diverses techniques de la mathématique discrète et de l'informatique afin de résoudre des problèmes d'optimisation combinatoire de la vie réelle [1].

Un problème d'optimisation combinatoire consiste à trouver la meilleure solution dans un ensemble discret de solutions appelé ensemble des solutions réalisables. En général, cet ensemble est fini mais de cardinalité très grande [1].

Il s'agit, en général, de maximiser (problème de maximisation) ou de minimiser (problème de minimisation) une fonction objectif sous certaines contraintes. Le but est de trouver la solution optimale dans un temps d'exécution raisonnable.

## 1.2 Problèmes d'optimisation combinatoire

### 1.2.1 Définition

L'optimisation combinatoire consiste à minimiser ou maximiser une fonction souvent appelée fonction coût, d'une ou plusieurs variables soumises à des contraintes. L'optimisation combinatoire occupe une place très importante en recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire [2]. Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles et ne possèdent donc pas à ce jour de solution algorithmique efficace valable pour toutes les données.

# Optimisation = modélisation + résolution

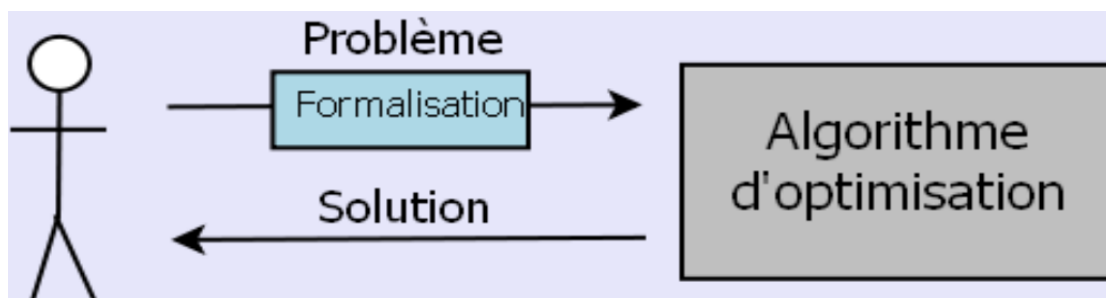


Figure 1.1 : L'optimisation combinatoire.

## 1.3 Exemples



Plusieurs problèmes existent dans la littérature qui décrivent les problèmes d'optimisation combinatoire parmi ces problèmes on a le problème de voyageur de commerce et le problème de sac à dos.

### 1.3.1 Le problème du voyageur de commerce :

En informatique, le problème du voyageur de commerce, ou « Travel Salesmen Problem », est un problème d'optimisation qui, étant donné une liste de villes, et des distances entre toutes les paires de villes, détermine un plus court chemin qui visite chaque ville une et une seule fois et qui termine dans la ville de départ.

Malgré la simplicité de son énoncé, il s'agit d'un problème d'optimisation pour lequel on ne connaît pas d'algorithme permettant de trouver une solution exacte rapidement dans tous les cas. Plus précisément, on ne connaît pas d'algorithme en temps polynomial, et sa version décisionnelle est un problème NP-complet, ce qui est un indice de sa difficulté. [1]



Figure 2 : Le problème de voyageur de commerce.

### 1.3.2 Le problème de Sac-à-dos

En algorithmique, le problème du sac à dos, noté également KP (en anglais, Knapsack problem) est un problème d'optimisation combinatoire. Il modélise une situation analogue au remplissage d'un sac à dos, ne pouvant supporter plus d'un certain poids, avec tout ou partie d'un ensemble donné d'objets ayant chacun un poids et une valeur. Les objets mis dans le sac à dos doivent maximiser la valeur totale, sans dépasser le poids maximum. [1]

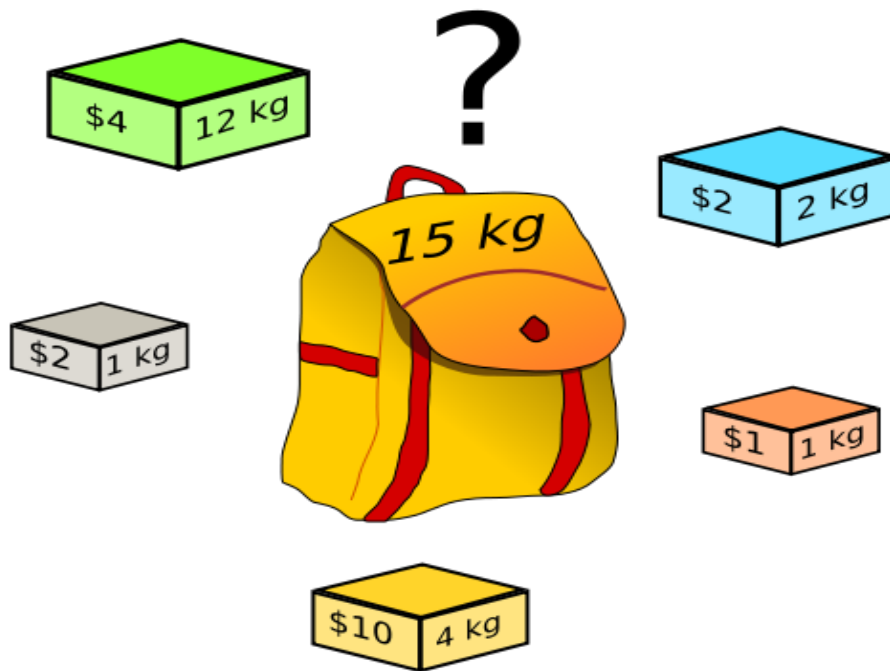


Figure 3 Le problème de Sac-à-dos

#### 1.4 Résolution d'un problème d'optimisation combinatoire

Résoudre un problème d'optimisation combinatoire nécessite l'étude de trois points particuliers :

- ❖ la définition de l'ensemble des solutions réalisables.
- ❖ l'expression de l'objectif à optimiser.
- ❖ le choix de la méthode d'optimisation à utiliser.

Les deux premiers points relèvent de la modélisation du problème, le troisième de sa résolution. Afin de définir l'ensemble des solutions réalisables, il est nécessaire d'exprimer l'ensemble des contraintes du problème. Ceci ne peut être fait qu'avec une bonne connaissance du problème sous étude et de son domaine d'application.

## 1.5 Les méthodes de résolution des problèmes d'optimisation combinatoire

Les méthodes d'optimisation peuvent être réparties en deux grandes classes de méthodes pour la résolution des problèmes :

- ❖ Les méthodes exactes.
- ❖ Les méthodes approchées.

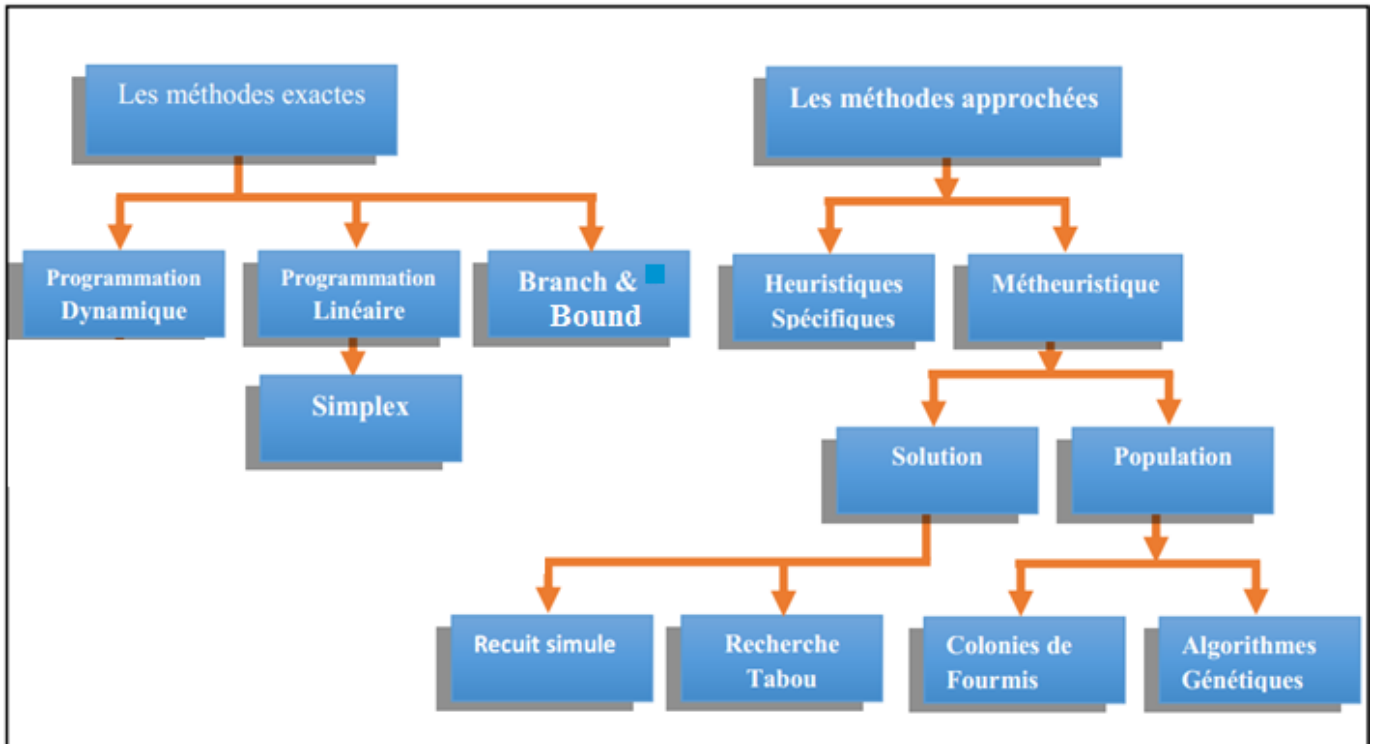


Figure 4 Classification des méthodes d'optimisation combinatoire

### 1.5.1 Les méthodes exactes

Une méthode de résolution est dite exacte si elle garantit l'obtention d'une solution optimale du problème. En effet, le temps de résolution de certains problèmes par les méthodes exactes connues augmente d'une manière exponentielle avec leurs tailles. Nous pouvons distinguer trois grands types de méthodes exactes :

- La programmation dynamique.
- Les méthodes de séparation et d'évaluation progressives (Branch and Bound).
- La méthode de programmation linéaire en nombre entière.

#### 1.5.1.1 La programmation dynamique

En informatique, la programmation dynamique est une méthode algorithmique pour résoudre des problèmes d'optimisation. Le concept a été introduit au début des années 1950 par Richard Bellman<sup>1</sup>. À l'époque, le terme « programmation » signifie planification et ordonnancement. La programmation dynamique consiste à résoudre un problème en le décomposant en sous-problèmes, puis à résoudre les sous-problèmes, des plus petits aux plus grands en stockant les résultats intermédiaires. Elle a d'emblée connu un grand succès, car de nombreuses fonctions économiques de l'industrie étaient de ce type. [2]

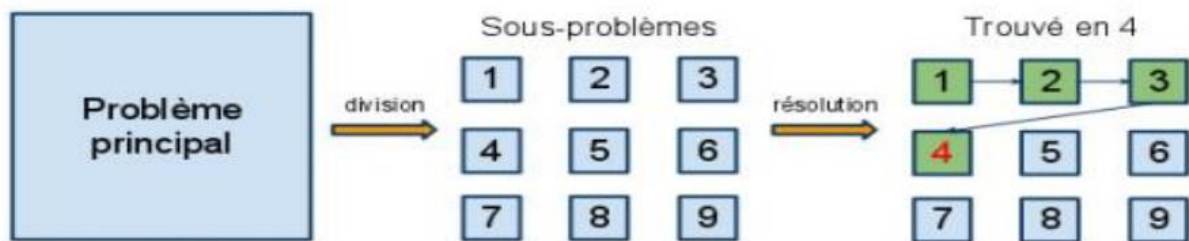


Figure 5 Divisé en sous-problèmes.

### 1.5.1.2 La méthode de simplexe

L'algorithme du simplexe est un algorithme de résolution des problèmes d'optimisation linéaire. Il a été introduit par George Dantzig à partir de 1947. C'est probablement le premier algorithme permettant de minimiser une fonction sur un ensemble défini par des inégalités. De ce fait, il a beaucoup contribué au démarrage de l'optimisation numérique. L'algorithme du simplexe a longtemps été la méthode la plus utilisée pour résoudre les problèmes d'optimisation linéaire. Depuis les années 1985 à 1990, il est concurrencé par les méthodes de points intérieurs, mais garde une place de choix dans certaines circonstances (en particulier si l'on a une idée des contraintes d'inégalité active en la solution). [2]

### 1.5.1.3 Branch and Bound

La technique du Branch & Bound est une méthode algorithmique classique pour résoudre un problème d'optimisation combinatoire. Il s'agit de rechercher une solution optimale dans un ensemble combinatoire de solutions possibles. La méthode repose d'abord sur la séparation (Branch) de l'ensemble des solutions en sous-ensembles plus petits. L'exploration de ces solutions utilise ensuite une évaluation optimiste pour majorer (bound) les sous-ensembles, ce qui permet de ne plus considérer que ceux susceptibles de contenir la solution potentiellement meilleure que la solution courante. [11]



## **1.5.2 Les méthodes approchées**

### **1.5.2.1 Heuristique**

Les heuristiques constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille en un temps de calcul très petit, une heuristique représente un bon compromis entre le temps de recherche et la qualité de solution. Feigenbaum et Feldman (1963) définissent une heuristique comme une règle d'estimation, une stratégie, une astuce, une simplification, ou toute autre sorte de système qui limite drastiquement la recherche des solutions dans l'espace des configurations possibles. [2]

### **1.5.2.2 Méta-Heuristique**

Une méta-heuristique est un algorithme d'optimisation visant à résoudre des problèmes d'optimisation difficiles (souvent issus des domaines de la recherche opérationnelle, de l'ingénierie ou de l'intelligence artificielle) pour lesquels on ne connaît pas de méthodes classiques plus efficace.

Les méta-heuristiques sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global, c'est-à-dire l'extremum global d'une fonction, par échantillonnage d'une fonction objectif. Elles se comportent comme des algorithmes de recherche, tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution (d'une manière proche des algorithmes d'approximation).

Les méta-heuristiques les plus connues sont :

- Les algorithmes évolutionnistes, parmi lesquels :
  1. les stratégies d'évolution.
  2. L'algorithme génétique.
  3. les algorithmes à évolution différentielle.
  4. les algorithmes à estimation de distribution.
  5. les systèmes immunitaires artificiels.
  6. la recombinaison de chemin (Path Relinking en anglais).
  
- le recuit simulé.
- L'algorithme de colonies de fourmis.
- Les algorithmes d'optimisation par essaims des particules.
- la recherche tabou.

- la méthode GRASP.

Il existe un très grand nombre d'autres méta-heuristiques, plus ou moins connues :

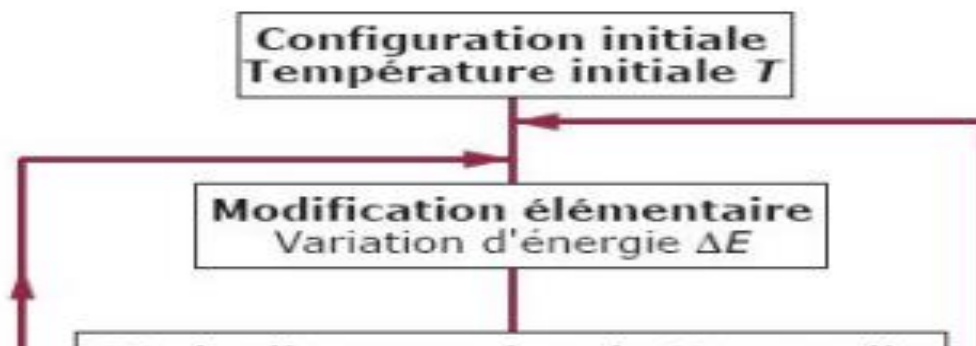
- ❖ L'algorithme du kangourou.
- ❖ La méthode de Fletcher et Powell.
- ❖ La méthode du bruitage.
- ❖ La tunnelisation stochastique.
- ❖ L'escalade de collines à recommencements aléatoires.
- ❖ La méthode de l'entropie croisée.
- ❖ L'algorithme de recherche d'harmonie.

La recherche dans le domaine étant très active, il est impossible de produire une liste exhaustive des différentes méta-heuristiques d'optimisation. La littérature spécialisée montre un grand nombre de variantes et d'hybridations entre méthodes, particulièrement dans le cas des algorithmes évolutionnaires.

### 1.5.2.2.1 La Recuit simulé

Le recuit simulé est une méthode empirique (méta-heuristique) inspirée d'un processus utilisé en métallurgie. On alterne dans cette dernière des cycles de refroidissement lent et de réchauffage (recuit) qui ont pour effet de minimiser l'énergie du matériau. Cette méthode est transposée en optimisation pour trouver les extrema d'une fonction.

La méthode vient du constat que le refroidissement naturel de certains métaux ne permet pas aux atomes de se placer dans la configuration la plus solide. La configuration la plus stable est atteinte en maîtrisant le refroidissement et en le ralentissant par un apport de chaleur externe, ou bien par une isolation.[14]



*Figure 6 Fonctionnement de l'algorithme de recuit simulé.*

#### **1.5.2.2.2 La recherche Tabou**

La recherche tabou (Tabou Search) est une méthode de recherche locale combinée avec un ensemble de techniques permettant d'éviter d'être piégé dans un minimum local ou la répétition d'un cycle. La recherche tabou est introduite principalement par Glover & Hansen 1986. Cette méthode a montré une grande efficacité pour la résolution des problèmes d'optimisation difficiles. [10]

#### **1.5.2.2.3 Les colonies de fourmis**

C'est une nouvelle méthode de résolution des problèmes combinatoires. Cette méta-heuristique a été proposée pour la première fois par Dorigo en 1992. Les colonies de fourmis

sont basées sur le comportement réel et la communication chez les fourmis. Il est connu que les fourmis sont capables de déterminer le chemin le plus court entre leur nid et une source de nourriture grâce à la phéromone déposés sur les trajets parcourus. Cela peut paraître surprenant au premier abord mais un chemin plus court reçoit plus de phéromones qu'un chemin plus long. Cet algorithme a été appliquée sur le problème du voyageur de commerce (TSP : Travelling Salesman problem).[12]

#### **1.5.2.2.4 La méthode GRASP :**

GRASP (Greedy Randomized Adaptive Search Procedure) est un algorithme metaheuristique et multi-start utilisé pour la résolution des problèmes d'optimisation combinatoire, chaque itération de cet algorithme se passe sur deux étapes, la première – étape de construction – essaie de construire une solution selon l'algorithme semi-glouton (semi-Greedy) et la deuxième – étape de recherche locale – essaie d'améliorer la solution construite dans la première étape, par une recherche locale. La solution finale sera la meilleure solution obtenue. [12]

#### **1.5.2.2.5 Les algorithmes génétiques**

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 des méthodes exactes (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. On va détailler cette méthode dans le chapitre suivant.

## **1.6 Conclusion**

Nous avons présenté dans ce chapitre plusieurs méthodes de résolution des problèmes combinatoires. La difficulté de ces problèmes nécessite de faire appel à des méthodes qui donnent des solutions pas forcément optimales mais proche de l'optimale, ce sont les méta-heuristiques, et parmi ces méthodes, il y a les algorithmes génétiques que nous avons choisi



comme méthode de résolution de notre problème. Le chapitre suivant décrit cette méthode et son efficacité.

# Les algorithmes génétiques

## Chapitre 2 :

## **2.1 Introduction**

Les algorithmes génétiques, initiés dans les années 1970 par John Holland, sont des algorithmes d'optimisation s'appuyant sur des techniques dérivées de la génétique et des mécanismes d'évolution de la nature : croisement, mutation, sélection. Dans un premier temps, nous nous attacherons à l'origine des algorithmes génétiques, leur appartenance dans le monde de la vie artificielle, et à la théorie de l'évolution des espèces formulée par le naturaliste Charles Darwin, sur laquelle ces premiers sont basés. [7]

## **2.2 Définition**

Les algorithmes génétiques est la variante la plus connue des algorithmes évolutionnaires, d'ailleurs, par abus de langage, beaucoup de spécialistes désignent et continuent à désigner les approches évolutionnaires par le terme « algorithmes génétiques ». Comme leur nom l'indique, les algorithmes génétiques s'inspirent du patrimoine héréditaire d'un individu (génotype) représenté par ses chromosomes. L'interaction du génotype d'un individu avec son environnement détermine son phénotype qui peut être modifié par mutation. Le phénotype est évalué par décodage du génotype, qui est une chaîne de symboles souvent binaires, dans le but de donner une valeur de performance exploitable par les opérateurs de sélection (couleur des yeux, des cheveux, traits du visage, etc.). [8]

## **2.3 Fonctionnement de l'algorithme génétique :**

Les opérateurs de variation (croisement et mutation) sont relatifs à la représentation binaire puisqu'ils agissent sur les chaînes binaires des génotypes. Dans un algorithme génétique simple, la recherche est réglée par l'application successive des opérateurs de variation.

Le croisement correspond à la phase de coopération entre les individus alors que la mutation correspond à la phase d'adaptation individuelle. On définit également la fonction coût (fitness), qui traduit la capacité d'adaptation des individus au problème considéré. La figure suivante représente le principe de fonctionnement de l'algorithme génétique :

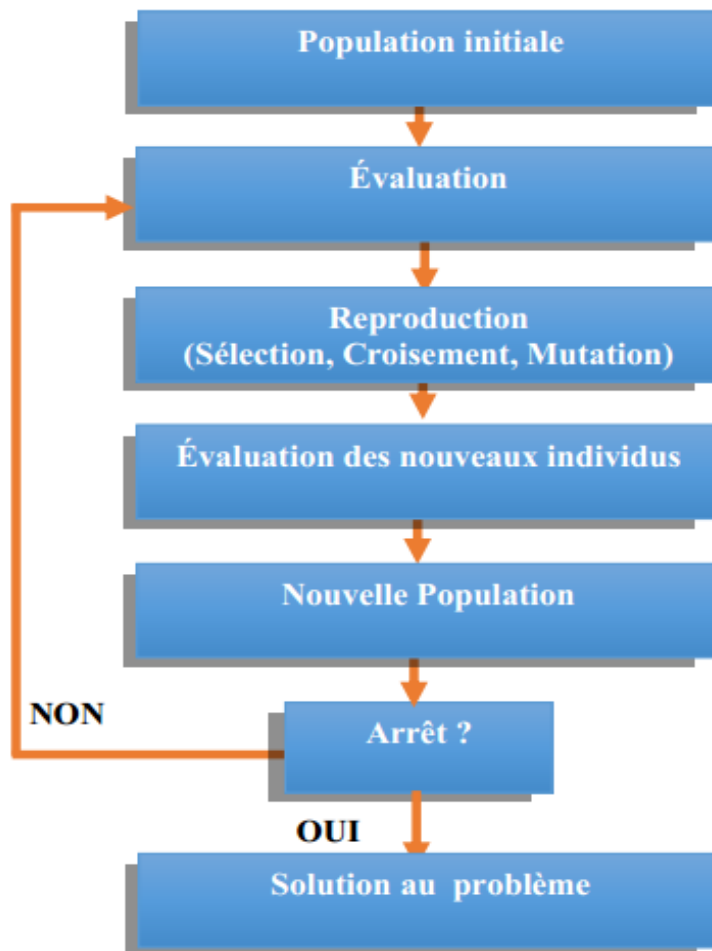


Figure 7 La procédure générale de l'algorithme génétique

Un algorithme génétique recherche le ou les extrema d'une fonction définie sur un espace de données. Pour l'utiliser, on doit disposer des cinq éléments suivants :

- 1- Un principe de codage de l'élément de population. Cette étape associe à chacun des points de l'espace d'état une structure de données. Elle se place généralement après une phase de modélisation mathématique du problème traité. La qualité du codage des données conditionne le succès des algorithmes génétiques. Les codages binaires ont été très utilisés à l'origine. Les codages réels sont désormais largement utilisés, notamment dans les domaines applicatifs pour l'optimisation de problèmes à variables réelles.
- 2- Un mécanisme de génération de la population initiale. Ce mécanisme doit être capable de produire une population d'individus non homogène qui servira de base pour les générations futures. Le choix de la population initiale est important car il peut rendre plus ou moins rapide la convergence vers l'optimum global. Dans le cas où l'on ne connaît rien du problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche.

- 3- Une fonction à optimiser. Celle-ci retourne une valeur appelée fitness ou fonction d'évaluation de l'individu.
- 4- Des opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace d'état. L'opérateur de croisement recompose les gènes d'individus existant dans la population, l'opérateur de mutation a pour but de garantir l'exploration de l'espace d'états.
- 5- Des paramètres de dimensionnement : la taille de la population, nombre total de générations ou critère d'arrêt, probabilités d'application des opérateurs de croisement et de mutation.

## 2.4 Structure générale de l'algorithme génétique

Un algorithme génétique générique à la forme suivante :

- 1) Initialiser la population initiale P.
- 2) Evaluer P.
- 3) **TantQue** (Pas Convergence) **faire** :
  - a) P ' = Sélection des Parents dans P
  - b) P ' = Appliquer Opérateur de Croisement sur P '
  - c) P ' = Appliquer Opérateur de Mutation sur P '
  - d) P = Remplacer les Anciens de P par leurs Descendants de P '
  - e) Evaluer P

**FinTantQue**

Le critère de convergence peut être de nature diverse, par exemple :

- Un taux minimum qu'on désire atteindre d'adaptation de la population au problème.
- Un certain temps de calcul à ne pas dépasser.
- Une combinaison de ces deux points.

## 2.5 Codage des individus



Le premier pas dans l'implantation des algorithmes génétiques est de créer une population d'individus initiaux. En effet, les algorithmes génétiques agissent sur une population d'individus, et non pas sur un individu isolé. Par analogie avec la biologie, chaque individu de la population est codé par un *chromosome* ou *génotype* (Holland, 1975). Une population est donc un ensemble de chromosomes. Chaque chromosome code un point de l'espace de recherche. L'efficacité de l'algorithme génétique va donc dépendre du choix du codage d'un chromosome.

Dans l'algorithme canonique de Holland (AGC), un chromosome était représenté sous forme de chaînes de bits contenant toute l'information nécessaire à la description d'un point dans l'espace ce qui permettait des opérateurs de mutations et de croisements simples



Figure 8 Codage des solutions.

Les algorithmes génétiques utilisent trois opérateurs pour générer de nouvelles solutions :

- ❖ L'opérateur de sélection qui permet de choisir des solutions parentes sur lesquelles la reproduction va être faite pour générer des nouvelles solutions.
- ❖ L'opérateur de croisement qui permet de croiser les deux solutions parentes et créer de nouvelles solutions.
- ❖ L'opérateur de mutation qui permet de diversifier les nouvelles solutions afin qu'elles ne ressemblent pas trop aux solutions parentes.

## 2.6 L'opérateur de sélection

La sélection consiste à choisir des individus qui permettront de générer de nouveaux individus. Plusieurs méthodes existent pour sélectionner des individus destinés à la reproduction. On trouve essentiellement quatre types de méthodes de sélection différentes :

- La méthode de la "loterie biaisée" (roulette wheel) de Goldberg.
- La méthode "élitiste".
- La sélection par tournois.
- La sélection universelle stochastique.

### a) La loterie biaisée (roulette wheel)

Cette méthode est la plus connue et la plus utilisée. La sélection des individus par le système de la roulette s'inspire des roues de loterie. À chacun des individus de la population est associé un secteur d'une roue. L'angle du secteur étant proportionnel à la qualité de l'individu qu'il représente. Vous tournez la roue et vous obtenez un individu. Les tirages des individus sont ainsi pondérés par leur qualité. Et presque logiquement, les meilleurs individus ont plus de chance d'être croisés et de participer à l'amélioration de notre population. (La figure 2.3) illustre une population de 5 individus dont les performances sont représentées en roulette. [8]

individu	Fonction objectif	pourcentage
1	30	39%
2	5	7%
3	16	21%
4	10	13%
5	15	15%

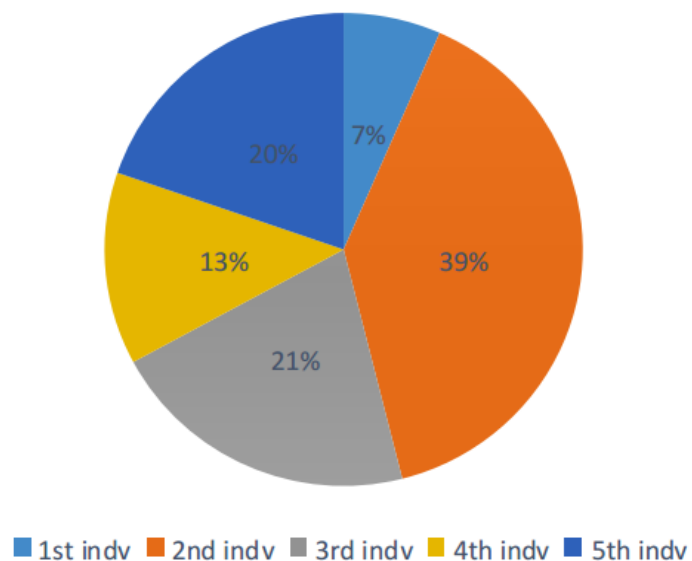


Figure 9 Sélection par roulette.

### b) La méthode élitiste

Cette méthode consiste à sélectionner les  $n$  individus dont on a besoin pour la nouvelle génération  $P$  en prenant les  $n$  meilleurs individus de la population  $P$  après l'avoir triée de manière décroissante selon la fitness de ses individus. Il est inutile de préciser que cette méthode est encore pire que celle de la loterie biaisée dans le sens où elle amènera à une convergence prématurée encore plus rapidement et surtout de manière encore plus sûre que la méthode de sélection de la loterie biaisée, en effet, la pression de la sélection est trop forte, la variance nulle et la diversité inexistante, au moins le peu de diversité qu'il pourrait y avoir ne résultera pas de la sélection mais plutôt du croisement et des mutations.

Là aussi il faut opter pour une autre méthode de sélection. [8]

### c) La sélection par tournoi

Le principe de la sélection par tournoi augmente les chances pour les individus de piètre qualité de participer à l'amélioration de la population. Le principe est très rapide à implémenter. Un tournoi consiste en une rencontre entre plusieurs individus pris au hasard dans la population. Le vainqueur du tournoi est l'individu de meilleure qualité. Cette méthode est en général satisfaisante (La figure 2.4). [8]

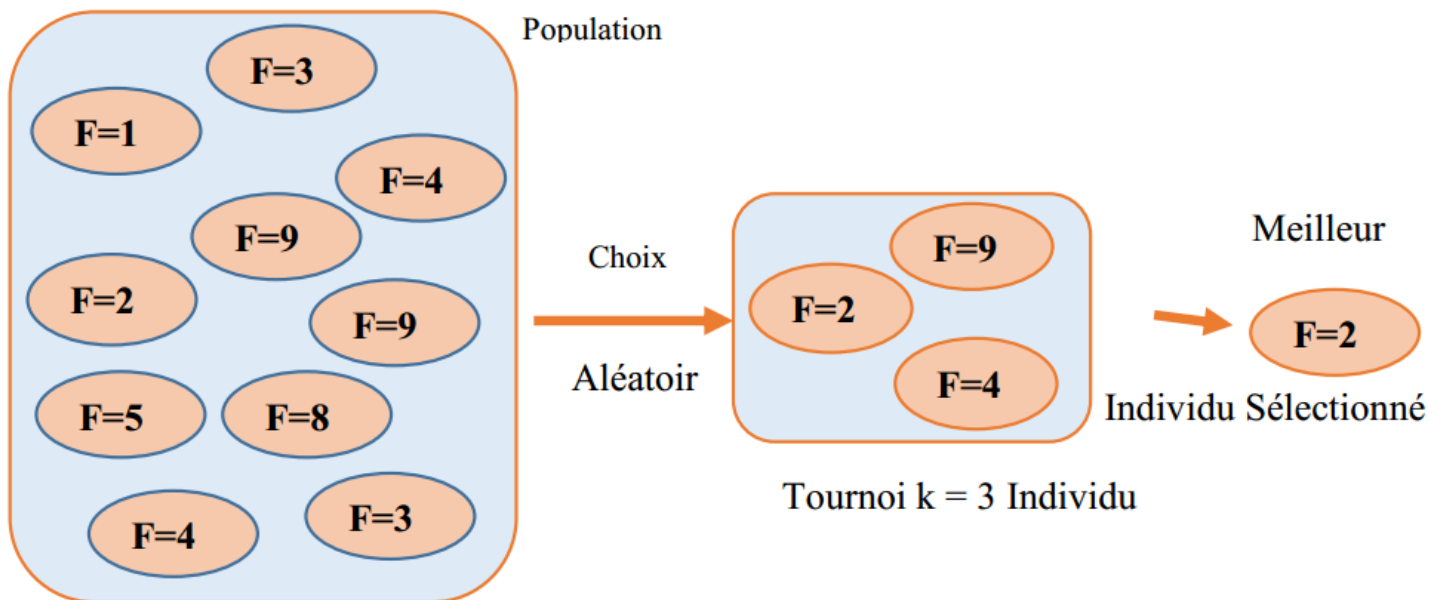


Figure 10 Sélection par tournoi.

### d) La sélection universelle stochastique

Cette méthode semble être très peu utilisée et qui plus est possède une variance faible, donc introduit peu de diversité, nous n'entrerons donc pas dans les détails, on se contentera d'exposer sa mise en œuvre : On prend l'image d'un segment découpe en autant de sous-segments qu'il y a d'individus. Les individus sélectionnés sont désignés par un ensemble de points équidistants. [8]

## 2.7 L'opérateur de croisement

Les croisements permettent de simuler des reproductions d'individus dans le but d'en créer des nouveaux. Il est tout à fait possible de faire des croisements aléatoires. Toutefois, une solution largement utilisée est d'effectuer des croisements multipoints.

En effet, plus le nombre de points de croisements sera grand et plus la probabilité de croisement sera élevée plus il y aura d'échange de segments, donc d'échange de paramètres, d'information, et plus le nombre de points de croisements sera petit et plus la probabilité de croisement sera faible, moins le croisement apportera de diversité. [8]

### i. Le croisement à un point

Il a été initialement défini pour le codage binaire. Le principe consiste à tirer aléatoire une position pour chaque parent et à échanger les sous-chaines des parents à partir des positions tirées. Ce qui donne naissance à deux nouveaux individus ind1 et ind2 (Figure 2.5).

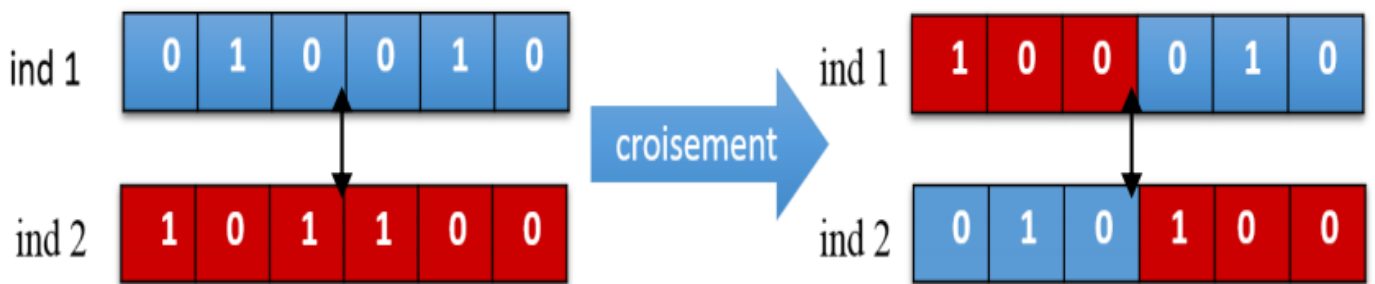


Figure 11 Croisement à un point.

### ii. Les croisements multipoints

Elle reprend le mécanisme de la méthode de croisement à un point en généralisant l'échange à 3 ou 4 sous chaines (Figure 2.6).

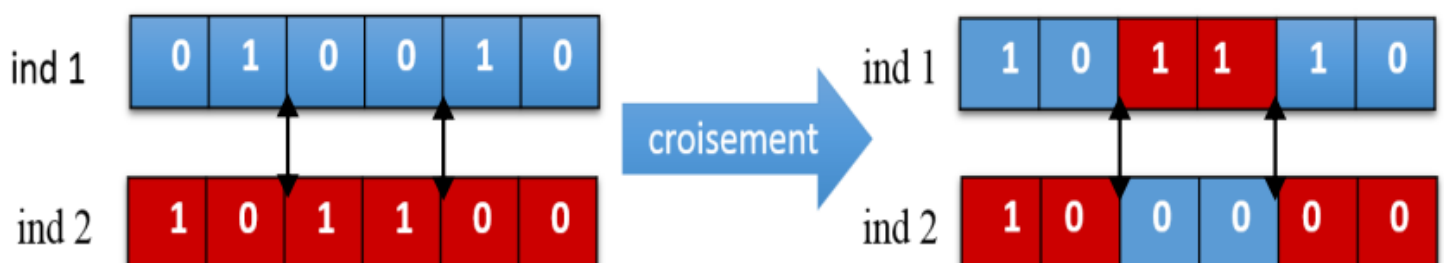


Figure 12 Croisement à deux points.

### iii. Le croisement uniforme

Pourquoi s'arrêter au croisement en deux points ? Pourquoi pas trois points ? Ou encore quatre ? Si on pousse cette démarche, on arrive au croisement uniforme. Le principe est le suivant : on prend deux parents, A et B ; pour chaque gènes on lance une pièce de monnaie ; si c'est face l'enfant prend le gène du parent A ; si c'est pile l'enfant prend le gène du parent B.

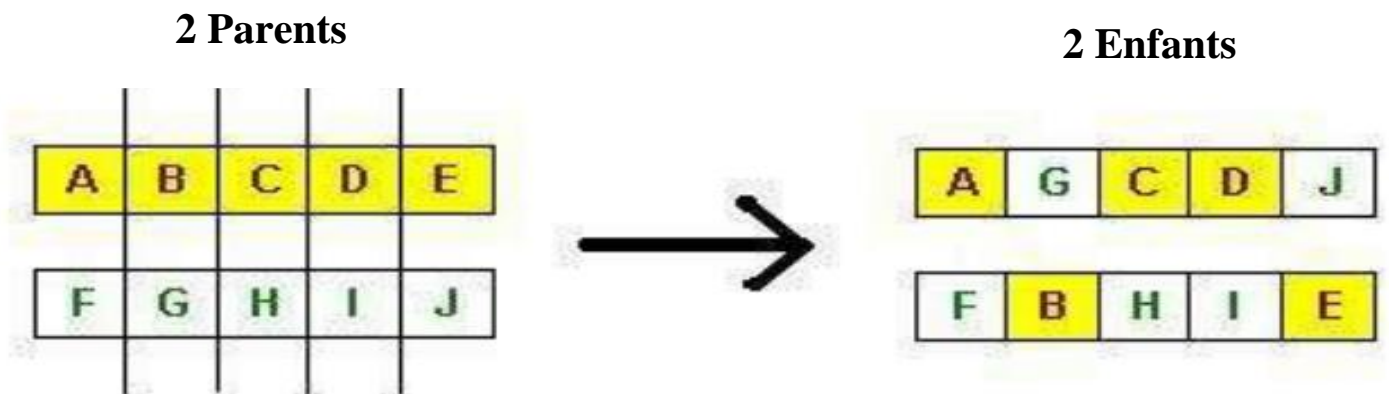


Figure 13 Croisement uniforme.

## 2.8 L'opérateur de mutation

L'opération de mutation protège les algorithmes génétiques des pertes prématurées 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 (Figure 2.8).

Une mutation consiste simplement en l'inversion d'un bit (ou de plusieurs bits, mais vu la probabilité de mutation c'est extrêmement rare) se trouvant en un locus bien particulier et lui aussi déterminé de manière aléatoire; on peut donc résumer la mutation de la façon suivante :

On utilise une fonction censée nous retourner True avec une probabilité  $pm$ .

**Pour chaque locus faire**

Faire appel à la fonction

**Si** cette fonction nous renvoie *true* **alors**

On inverse le bit se trouvant à ce locus

**FinSi**

**FinPour ;**



Figure 14 Une mutation.

L'opérateur de mutation modifie donc de manière complètement aléatoire les caractéristiques d'une solution, ce qui permet d'introduire et de maintenir la diversité au sein de notre population de solutions. Cet opérateur joue le rôle d'un "élément perturbateur", il introduit du "bruit" au sein de la population.

Cet opérateur dispose de 4 grands avantages :

- ❖ Il garantit la diversité de la population, ce qui est primordial pour les algorithmes génétiques.
- ❖ Il permet d'éviter un phénomène connu sous le nom de dérive génétique. On parle de dérive génétique quand certains gènes favorisés par le hasard se répandent au détriment des autres et sont ainsi présents au même endroit sur tous les chromosomes. Le fait que l'opérateur de mutation puisse entraîner de manière aléatoire des changements au niveau de n'importe quel locus permet d'éviter l'installation de cette situation défavorable.
- ❖ Il permet de limiter les risques d'une convergence prématurée causée par exemple par une méthode de sélection élitiste imposant à la population une pression sélective trop forte. En effet, dans le cas d'une convergence prématurée on se retrouve avec une population dont tous les individus sont identiques mais ne sont que des optimums locaux.
- ❖ La mutation permet d'atteindre la propriété d'ergodicité.

L'ergodicité est une propriété garantissant que chaque point de l'espace de recherche puisse être atteint.

## 2.9 L'opérateur de remplacement

Cet opérateur est consisté à réintroduire les descendants obtenus par application successive des opérateurs de sélection, de croisement et de mutation (la population P') dans la population de leurs parents (la population P). On trouve essentiellement 2 méthodes de remplacement différentes :

### a. Le remplacement stationnaire

Dans ce cas, les enfants remplacent automatiquement les parents sans tenir compte de leurs performances respectives, et le nombre d'individus de la population ne varie pas tout au long du cycle d'évolution simulé, ce qui implique donc d'initialiser la population initiale avec un nombre suffisant d'individus. Cette méthode peut être mise en œuvre de 2 façons différentes :

- La première se contente de remplacer la totalité de la population P par la population P', cette méthode est connue sous le nom de remplacement générationnel.
  
- La deuxième méthode consiste à choisir une certaine proportion d'individus de P' qui remplaceront leurs parents dans P (proportion égale à 100 % dans le cas du remplacement générationnel).

### b. Le remplacement élitiste

Dans ce cas, on garde au moins l'individu possédant les meilleures performances d'une génération à la suivante. En général, on peut partir du principe qu'un nouvel individu (enfant) prend place au sein de la population que s'il remplit le critère d'être plus performant que le moins performant des individus de la population précédente. Donc les enfants d'une génération ne remplaceront pas nécessairement leurs parents comme dans le remplacement stationnaire et par la même la taille de la population n'est pas figée au cours du temps.

## 2.10 Conclusion

Nous avons présenté au cours de ce chapitre l'algorithme génétique. Il est nécessaire de faire appel à des heuristiques permettant de trouver de bonnes solutions approchées.

Après la présentation du problème qu'on va traiter dans le chapitre suivant on va choisir certaines méthodes adéquates vues dans ce chapitre pour la résolution de notre problème.



# Chapitre 3

# **Étude de cas : Problème des interventions du service technique de la société SONELGAZ**

### **3.1 Introduction**

Nous nous intéressons dans ce chapitre au problème des interventions des groupes techniques de la société de SONELGAZ. L'objectif est de minimiser la durée totale de réalisation de ces demandes en favorisant la qualité de services. L'approche de résolution fait appel aux algorithmes génétiques, développé dans le chapitre précédent.

On va commencer par une présentation de la société de SONELGAZ, ensuite on va faire une étude du problème traité, sa complexité, et sa formulation mathématique adéquat, et on va terminer par une description de la solution proposée.

### **3.2 Présentation de la société SONELGAZ**

SONELGAZ, ou Société Nationale de l'Electricité et du Gaz, est une compagnie chargée de la production, du transport et de la distribution de l'électricité et du gaz en Algérie.

SONELGAZ est aujourd'hui érigé en Groupe industriel composé de 39 filiales et 5 sociétés en participation. Ainsi, ses filiales métiers de base assurent la production, le transport et la distribution de l'électricité ainsi que le transport et la distribution du gaz par canalisations.

On compte :

- La Société de Production de l'Electricité (SPE),
- Sharikat Kahraba wa takat moutadjadida (SKTM)
- La Société de l'Engineering de l'électricité et du Gaz (CEEG)
- La Société de Gestion du Réseau de Transport de l'Electricité (GRTE),
- La Société de Gestion du Réseau de Transport Gaz (GRTG),
- L'Opérateur Système électrique (OS), chargée de la conduite du système Production / Transport de l'électricité,
- La Société de Distribution de l'électricité et du gaz d'Alger (SDA),
- La Société de Distribution de l'électricité et du gaz du Centre (SDC),
- La Société Distribution de l'électricité et du gaz de l'Est (SDE),
- La Société de Distribution de l'électricité et du gaz de l'Ouest (SDO).

### **3.3 La Direction de Distribution de Mila**

La société a une agence régionale au niveau de la wilaya de Mila. Cette agence se compose de plusieurs annexes commerciales répartis sur 4 communes de la wilaya.

**Siège social :** Avenue de l'ALN - Mila

**Tél :** 031.46.38.51/52

**Fax :** 031.57.51.16

**Couverture territoriale :** Wilaya de Mila

**Nombre de communes :** 32

**Nombre de clients Electricité (BT/MT/HT) :** 154 672

**Nombre de clients Gaz (BP/MP/HP) :** 81 339

### **Les agences commerciales**

- Mila (Communes rattachées : Mila, Grarem Gouga, Sidi Merouane, Hamala, Ain Tine,
- Chelghoum Laid (Communes rattachées : Ch-Laid, Oued Athmania, Ain Melouk, Teleghma, Oued Seguen et MChira)
- Ferdjioua (Communes rattachées : Ferdjioua, Oued Enedja, Ahmed Rachedi, Tiberguent, Bouhatem, Rouached, Tassala, Tassadane, Derrahi Bouslah, Zerraza, Amira Arres, Tarrai Beinen, Zeghaia, El Ayadi Barbesse, Ain Baida Ahriche et Yahia Beni Guechaet)
- Tadjnanaet (Communes rattachées : Tadjenanet, Benyahia Abderrahmene et Ouled Khlouf.).

Chaque centre contient un ou plusieurs groupes techniques qui se chargent de répondre aux demandes des citoyens.

### **3.4 Présentation du problème**

Le problème traité dans ce travail concerne les interventions des groupes techniques de la société de SONELGAZ qui surviennent suite aux requêtes qui arrivent au centre de l'agence depuis les citoyens de la région, ces requêtes sont de multiples natures :

- Nouveau raccordement,
- Coupures d'électricité,
- Fuite du gaz,
- Demande d'extension du réseau,
- Défiance du compteur,...

Chaque type de requêtes nécessite un groupe particulier de service technique avec des équipements matériels particuliers.

La résolution de ce problème consiste à trouver un plan d'affectation des requêtes aux groupes techniques de façon à minimiser le temps de réponse aux requêtes et à maximiser le rendement des groupes technique en respectant les contraintes de disponibilité de ressources matérielles et de la durée de travail des groupes par jour.

Ce problème peut être assimilé à un problème de transport, qui consiste à trouver un ensemble de tournées permettant de visiter un ensemble de cites de façon à minimiser la distance totale parcourue en respectant certains contraintes. Ce problème est connu dans la littérature sous le nom de « problème de tournées de véhicule » qui a été traité depuis plus de trente ans. Les chercheurs ont définis plusieurs variantes de ce problème parmi eux on a le problème de transport à la demande ou en anglais « Dial A Ride Problem ».

### 3.5 Le problème de Transport A la Demande (TAD)

Lors de ces dernières années, l'offre de transport public n'a cessé de se développer et de s'améliorer, et avec elle les demandes et les exigences des usages, rendant ces moyens de transport moins compétitifs surtout dans les milieux ruraux et périurbains. L'utilisation de la voiture individuel reste assez répondu dans ces zones de par sa disponibilité et de son efficacité malgré les quelques inconvénients qu'elle possède en l'occurrence le coût d'utilisation (prix de l'essence) et les problèmes environnementaux qu'elle cause.

L'une des solutions qui peut résoudre ce problème et permet de profiter des avantages des deux modes de transport (en commun et individuel) et de réduire leurs inconvénients est partage des voitures privées. Le service de partage des voitures privées est appelé service de Transport À la Demande (TAD). Dans ce mode de transport, les clients (voyageurs) envoient une demande de transport à un opérateur comprenant le lieu de ramassage (origine), la destination finale, le nombre de personnes à transporter et l'heure de passage désiré. Le problème de TAD consiste à élaborer des tournées de véhicules pour répondre aux demandes de transport en minimisant les coûts engendrés par ces tournées (distance parcourue, temps de service, nombre de véhicules utilisés) et en fournissant une bonne qualité de service aux voyageurs (temps de voyage, nombre de stations visitées, etc..) [1]

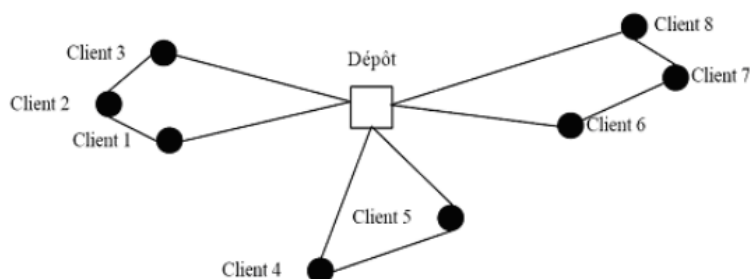


Figure 15 Exemple de problème à la demande

La résolution de notre problème consiste à trouver des tournées de véhicules de service technique de la société pour répondre aux requêtes des citoyens en minimisant la distance totale parcourue et en fournissant une bonne qualité de services.

Ce type de problèmes est qualifié dans la littérature parmi les problèmes dites NP-difficile ; c'est-à-dire il n'existe pas des méthodes exactes qui assurent la solution optimale en un temps raisonnable pour ses différentes dimensions. Pour cela il y a un recours aux méthodes approchées assurent des solutions proches de l'optimale en un temps raisonnable.

Pour résoudre notre problème on a choisi l'algorithme génétique ; c'est une méta-heuristique approchée qui a prouvé son efficacité dans la littérature pour ce type de problème.

Dans ce qui suit, on va présenter notre proposition en détaillant les techniques utilisées dans chaque partie de l'algorithme génétique développé.

### **3.6 L'algorithme génétique développé**

Cette technique part du principe évolutif de la sélection naturelle de Darwin. Celle-ci énonçait que les individus les plus aptes à survivre (les « meilleurs ») se reproduiront plus souvent et auront plus de descendants. Ainsi, la qualité du pool génétique de la population sera augmentée, les gènes plus efficaces deviendront plus fréquents ; la population s'améliore. Selon le même principe, un algorithme génétique part d'une population de solutions initiales, les fait se reproduire (les meilleures solutions ont plus de chances de se reproduire), créant ainsi la nouvelle génération de solutions. En répétant ce cycle plusieurs fois, on obtient une population composée de solutions meilleures. On utilise généralement les algorithmes génétiques pour trouver une solution, la meilleure solution après un certain nombre de générations.

Voici le corps principal d'une itération d'un algorithme génétique :

- 1- Évaluer la qualité (*fitness*) des individus et leurs chances de survie
- 2- Sélectionner les individus pour la reproduction
- 3- Effectuer la reproduction
- 4- Remplacer l'ancienne population par la nouvelle

Cette itération est répétée autant de fois que demandé. L'évaluation de la qualité (fitness) d'un individu permet d'illustrer avec une valeur numérique (et donc plus facile à manipuler) la qualité des gènes qui forment l'individu. Plus la qualité d'un individu est grande, plus ses chances d'être sélectionné pour la reproduction sont grandes. En général, on calcule les chances de reproduction d'un individu en regard de sa qualité en relation avec la qualité totale des individus de la population. La reproduction se fait en croisant deux individus. On applique sur les deux individus choisis des opérateurs génériques, habituellement l'enjambement (*cross-over*) et la mutation. La reproduction donne deux enfants (*offspring*), qui sont placés dans la nouvelle population. On répète la reproduction jusqu'à ce qu'on ait rempli la nouvelle population (la taille de la population devrait rester constante). On remplace alors l'ancienne population par la nouvelle, et on recommence le processus le nombre de générations voulu.

Voici un algorithme plus détaillé d'un algorithme génétique :

#### **Algorithme génétique générique**

- 1- Générer une population de individus de taille  $N$  :  $x_1, x_2, x_3, \dots, x_N$ .
- 2- Calculer les chances de survie (qualité ou encore *fitness*) de chaque individu.  $f(x_1), f(x_2), f(x_3), \dots, f(x_N)$ .
- 3- Vérifier si le critère de terminaison est atteint. Si oui, terminer.
- 4- Choisir une paire d'individus pour la reproduction (selon les chances de survie de chaque individu).
- 5- Selon les probabilités associées à chaque opérateur génétique, appliquer ces opérateurs.
- 6- Placer les individus produits dans la nouvelle population.
- 7- Vérifier si la taille de la nouvelle population est correcte. Si non, retourner à l'étape 4.
- 8- Remplacer l'ancienne population d'individus par la nouvelle.
- 9- Retourner à l'étape 2.

#### ➤ **Le codage de l'individu**



Les individus correspondent aux « solutions » du problème à optimiser. Dans notre cas l'individu est une suite de numéros qui représente l'ordre de satisfaction des demandes des interventions des citoyens commencé et terminé par '0' qui représente le centre de la société. Puisque on ne peut pas répondre aux demandes par une seule tournée, l'individu est découpé en tournées par des '0', donc entre deux '0' successives il y a une tournée qui doit être réalisée par un groupe de service technique. La figure suivante représente un individu qui contient 6 requêtes regroupées en 2 tournées.

0	3	1	5	0	2	4	6	0
---	---	---	---	---	---	---	---	---

Figure 16 Exemple de codage.

### ➤ La population initiale

La population initiale est générée par une méthode d'insertion aléatoire, en cherchant à favoriser la diversité de la population pour mieux explorer l'espace des solutions.

Cette approche pose deux problèmes essentiels ; le premier concerne la génération des individus invalides, vu que les solutions doivent satisfaire les contraintes du problème, à savoir la satisfaction de toutes les requêtes, le deuxième concerne la génération des individus irréalisable, vu que la tournée ne doit pas dépasser un temps maximal.

Pour résoudre le problème de redondance des requêtes et le problème de satisfaction de toutes les requêtes, on a modifié la méthode d'insertion aléatoire en une méthode aléatoire guidée, c'est-à-dire générer aléatoirement les individus en respectant ces deux contraintes.

Et pour le problème de tournées irréalisables, on a développé un algorithme s'appelle « Split », il prend l'individu composé d'une seule tournée et le découpe en tournées réalisables.

L'algorithme suivant illustre le principe de Split :

---

### Algorithme 4.3 : Le principe de Split

---

```

1 : Début
2 :   initialise le compteur de la durée maximal d'une tournée ;
3 :   // démarrer à partir de la première requête de l'individu
4 :   Tant que (n'est pas la fin de l'individu) Faire
5 :     initialise le compteur de la durée maximal d'une tournée ;
6 :     Tant que (il reste du temps pour ajouter une requête à la tournée) Faire
7 :       Ajouter la requête en cours à la tournée
8 :       Mettre à jour le compteur de la tournée
9 :       Passez à la requête suivante
10 :    Fin tant que
11 :    mettez le délimiteur de la tournée // le numéro '0'
12 :    Passez à la requête suivante
13 :  Fin tant que
14 :  Fin.

```

#### ➤ **Mesure de la qualité (*fitness*) d'un individu**

Le calcul de la qualité d'un individu est essentiel aux algorithmes génétiques. Cette fonction donne, en valeur numérique (habituellement réelle), la qualité d'un individu. La fonction de *fitness* dans notre cas représente la somme des distances entre les villes des demandes selon un chemin spécifié par la fonction Split.

#### ➤ **La sélection**

La méthode de sélection élaborée est une combinaison de deux techniques de sélection, la méthode stochastique et la méthode élitiste, et cela en commençant par l'application de la méthode élitiste sur  $\frac{1}{4}$  de la population puis on applique la méthode stochastique sur les  $\frac{3}{4}$  restant de la population, la méthode élitiste est appliquée selon la qualité des individus (le *fitness*).

Cette combinaison assure que les meilleurs individus de la population courante seront apparus dans la prochaine population et donne aux autres individus une possibilité d'être aussi dans la population. Le pseudo code de la sélection est donné dans l'algorithme suivant :

---

#### **Algorithme 4.3 : Le principe de Sélection**

```

1 : Début

```

- 2 : sélectionner  $\frac{1}{4}$  les meilleurs de la population ;
  - 3 : sélectionner  $\frac{1}{4}$  aléatoire de la reste de la population ;
  - 4 : **Fin.**
- 

### ➤ Le croisement

Pour le croisement nous avons utilisé la méthode à un seul point de coupure. Le principe consiste à choisir aléatoirement une position pour chaque parent et à échanger les sous-chainés des parents à partir des positions choisies.

### Algorithme 4.3 : Le principe de Croisement

---

- 1 : **Début**
  - 2 : sélectionner 2 individus parmi la liste sélectionné ;
  - 3 : générer un nombre aléatoire entre 0 et 100 ;
  - 4 : **si** (ce nombre est inférieur au Probabilité de croisement) **Alors**
  - 5 : | générer aléatoire un point de coupure ;
  - 6 : | concaténer la première partie de P1 avec la deuxième partie de P2
  - 7 : | concaténer la première partie de P2 avec la deuxième partie de P1
  - 8 : | vérifier l'existence d'une répétition ;
  - 9 : | ajouter les pères et ses fils à la population ;
  - 10 : **Sinon**
  - 11 : | ajouter seulement les pères à la population ;
  - 13 : **Finsi**
  - 14 : **Fin.**
- 

### ➤ La mutation

L'opérateur de mutation, bien qu'ayant une probabilité bien moindre (habituellement entre 0,5% et 5%) joue un rôle très important. Une reproduction utilisant uniquement le croisement est une méthode de qui est limitée par l'atteinte de maxima locaux. En effet, les gènes des enfants sont limités par les gènes des parents, et si un gène n'est pas présent dans la population initiale (ou s'il disparaît à cause des reproductions), il ne pourra jamais se développer chez les descendants. L'opérateur de mutation est là pour contourner ce problème. Chaque gène possède une faible probabilité de muter, c'est-à-dire d'être

aléatoirement remplacé par une autre incarnation de ce gène. Cette précaution permet de conserver ce qu'on appelle la diversité génétique. Dans notre cas, on applique l'opérateur de mutation en appliquant une mutation point à point et cela est fait par la permutation des deux gènes consécutifs.

---

#### Algorithme 4.3 : Le principe de Split

---

```
1 : Début
2 :   générer un nombre aléatoire entre 0 et 100 ;
3 :   si (ce nombre est inférieur au Probabilité de mutation) Alors
4 :     permuter entre les deux chromosomes qui se trouvent à la position n et n+1
5 :   Fin si
   Fin.
```

---

#### ➤ La condition d'arrêt

Généralement, un algorithme génétique se termine après un certain nombre de générations.

---

```
1 : Début
2 :   générer la population initiale d'une façon aléatoire ;
3 :   Tant que (i inférieur a nombre de génération maximal) Faire
4 :     sélectionner 1 /2 de la population ;
5 :     applique l'étape de croisement ;
6 :     Applique l'étape de mutation ;
7 :     Choisir le meilleur individu ;
8 :     Incréments i ;
9 :   Fin tant que
10 : Fin.
```

---

### **3.7 Conclusion**

Dans ce chapitre on a décrit la conception de notre algorithme génétique. tout d'abord on a présenté le problème réel traité qui est le problème des interventions des équipes techniques de la société de SONELGAZ, en suite on a fait la projection du problème sur le problème de transport à la demande qui est très connu dans le domaine de recherche opérationnelle, et qui est qualifié des problèmes NP-difficile, et on a terminé par une présentation des techniques utilisées dans chaque étape de l'algorithme génétique. Le chapitre suivant donne une discussion des résultats obtenus par l'application de cet algorithme sur des données réelles.

# **Chapitre 04 :**

# Implémentation et analyse expérimentale

## 4.1 Introduction

Dans ce dernier chapitre, nous présentons les différents résultats issus de l'application de l'algorithme génétique développé sur les données accumulées à partir des centres de la société de SONELGAZ au niveau de la wilaya de Mila, ensuite on va faire une comparaison avec la réalité appliquée. Pour mieux analyser les résultats, on va présenter les outils logiciels utilisés pour l'implémentation et les différentes interfaces de l'application.

## 4.2 Plateforme logicielle

Nom Version	et	Description	Logo
----------------	----	-------------	------




<b>Eclipse Luna M2</b>	Permet le codage et l'implémentation des classes java.	
<b>Xampp</b>	Support pour phpMyadmin et MySQL.	
<b>PhpMyAdmin 3.4.10.1</b>	Permet la gestion et la manipulation des bases de données MySQL	

Tableau 1: plateforme de travail.

### 4.3 Langage JAVA :

Java est un langage de programmation moderne développé par Sun Microsystems (aujourd'hui racheté par Oracle). Ce langage reprend en grande partie la syntaxe du langage C++, très utilisé par les informaticiens.

Notre choix est porté sur ce langage à cause des caractéristiques suivantes :

- **JAVA** est orienté objets et simple ce qui réduit le risque d'erreurs et d'incohérence.
- **JAVA** possède une bibliothèque immense de classes prêtes à l'emploi, ce qui facilite pleinement la procédure d'implémentation.
- Une des plus grandes forces de ce langage est son excellente portabilité, il permet aux concepteurs, selon le principe « write once, run every where », d'exécuter leurs applications sur tout environnement doté d'un JRE (Java Runtime Environnement - Environnement d'exécution java).

### 4.4 SGBD MySQL

Le **SGBD MySQL** est un serveur de relationnelles SQL. Il permet de gérer des bases de données, c'est à dire il est capable d'enregistrer, modifier et rechercher rapidement des données.

On a besoin d'une base de données pour sauvegarder les informations concernant les demandes des citoyens ; le numéro de la demande, nom, prénom, et adresse du citoyen, et une description de sa demande.

On a besoin d'une table qui contient toutes les communes ou villes de la wilaya de Mila et les distances entre eux.



id	numero	Nom
----	--------	-----

Figure4. 17 Table Ville de la base de données



+ Options			id	numero	Nom				
<input type="checkbox"/>		Modifier		Copier		Effacer	23	0	Centre Mila
<input type="checkbox"/>		Modifier		Copier		Effacer	24	1	Mila
<input type="checkbox"/>		Modifier		Copier		Effacer	25	2	Oued athmenia
<input type="checkbox"/>		Modifier		Copier		Effacer	26	3	Cheighoum laid
<input type="checkbox"/>		Modifier		Copier		Effacer	27	4	farjiwa
<input type="checkbox"/>		Modifier		Copier		Effacer	28	5	zaghaya
<input type="checkbox"/>		Modifier		Copier		Effacer	29	6	Redjas
<input type="checkbox"/>		Modifier		Copier		Effacer	30	7	tadjenanet
<input type="checkbox"/>		Modifier		Copier		Effacer	31	8	Ahmed Rashidi
<input type="checkbox"/>		Modifier		Copier		Effacer	32	9	aWlad Khallouf
<input type="checkbox"/>		Modifier		Copier		Effacer	33	10	Ben Yahia
<input type="checkbox"/>		Modifier		Copier		Effacer	34	11	Bouhatem
<input type="checkbox"/>		Modifier		Copier		Effacer	35	12	Tiberguent
<input type="checkbox"/>		Modifier		Copier		Effacer	36	13	Tessala
<input type="checkbox"/>		Modifier		Copier		Effacer	37	14	Tassadane
<input type="checkbox"/>		Modifier		Copier		Effacer	38	15	Rouached
<input type="checkbox"/>		Modifier		Copier		Effacer	39	16	Sidi Khalifa
<input type="checkbox"/>		Modifier		Copier		Effacer	40	17	Sidi Marwan
<input type="checkbox"/>		Modifier		Copier		Effacer	41	18	Chigara
<input type="checkbox"/>		Modifier		Copier		Effacer	42	19	Ain tin
<input type="checkbox"/>		Modifier		Copier		Effacer	43	20	Ain Imlok
<input type="checkbox"/>		Modifier		Copier		Effacer	44	21	Grarem

Figure4. 18 Liste des villes de la Wilaya de Mila

La table Distance représente la distance entre une ville source et une ville destination et contient numéro de la ville source, numéro de la ville destination et la distance entre les deux

dbville distance	
	id : int(11)
#	numSource : int(11)
#	numDestination : int(11)
#	distance : int(11)

Figure 4.19 Table Distance de la base de données

## 4.5 Description des interfaces de l'application

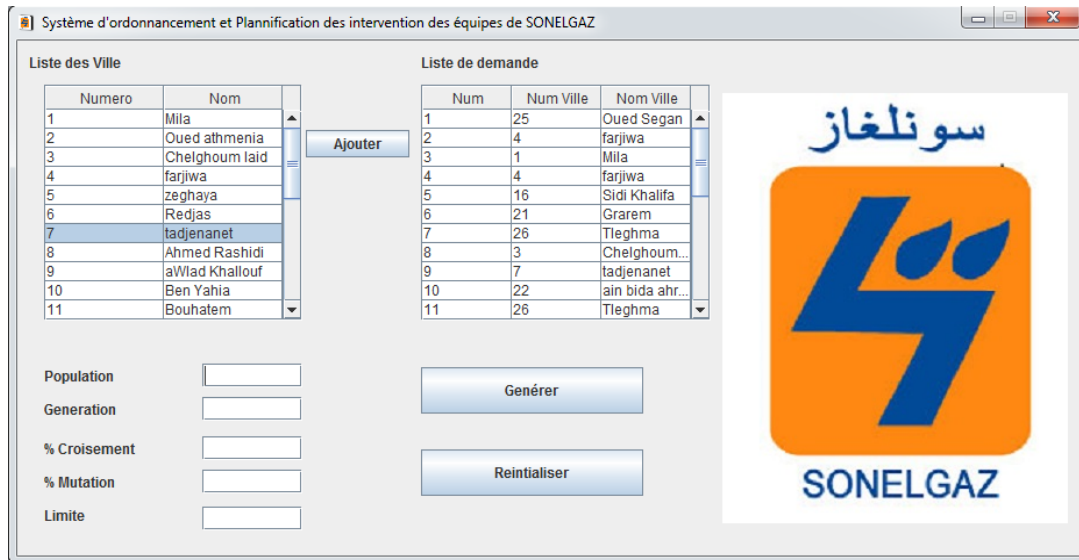


Figure 4.20 la fenêtre d'accueil de l'application

## 4.6 La fenêtre d'accueil de l'application

Pour simplifier les choses, on a séparé la partie base de données de la partie traitement de l'algorithme génétique, pour ce dernier, l'ajout d'une demande consiste à sélectionner la ville de la source de la demande dans une liste qui contient les villes de la wilaya de Mila puis cliquer sur le bouton Ajouter, la demande sera ajouté automatiquement avec un numéro unique (figure 5).

Numero	Nom
1	Mila
2	Oued athmenia
3	Chelghoum laid
4	farjiwa
5	zaghaya
6	Redjas
7	tadjenanet
8	Ahmed Rashidi
9	aWlad Khallouf
10	Ben Yahia
11	Bouhatem

**Ajouter**

Num	Num Ville	Nom Ville

Figure 4.21 Ajout de demandes

Après le remplissage de la liste de demandes, l'utilisateur doit saisir les données nécessaires à l'algorithme génétique, à savoir la taille de population, le nombre de génération, la probabilité de croisement et de mutation, et Limite qui représente la durée maximale d'une tournée (figure 6)

**Population**

**Generation**

**% Croisement**

**% Mutation**

**Limite**

**Générer**

**Reinitialiser**

Figure 4.22 Formulaire de saisie de paramètres de l'algorithme génétique.

Pour avoir les résultats l'utilisateur doit cliquer sur le bouton Générer, une fenêtre s'ouvre en affichant les évolutions de calcul et affiche le chemin entre les villes dont les demandes sont traitées comme indique la figure ci-dessous. Le traitement est lancé en cliquant sur le bouton Générer, une fenêtre s'ouvre en affichant l'évolution du résultat obtenu, génération par génération. Le résultat est affiché en forme textuelle accompagné d'une carte qui simule la distribution géographique des villes de la wilaya de Mila avec un graphe qui représente les tournée satisfaisant les demandes des citoyens, la figure 7 donne un exemple d'une solution.

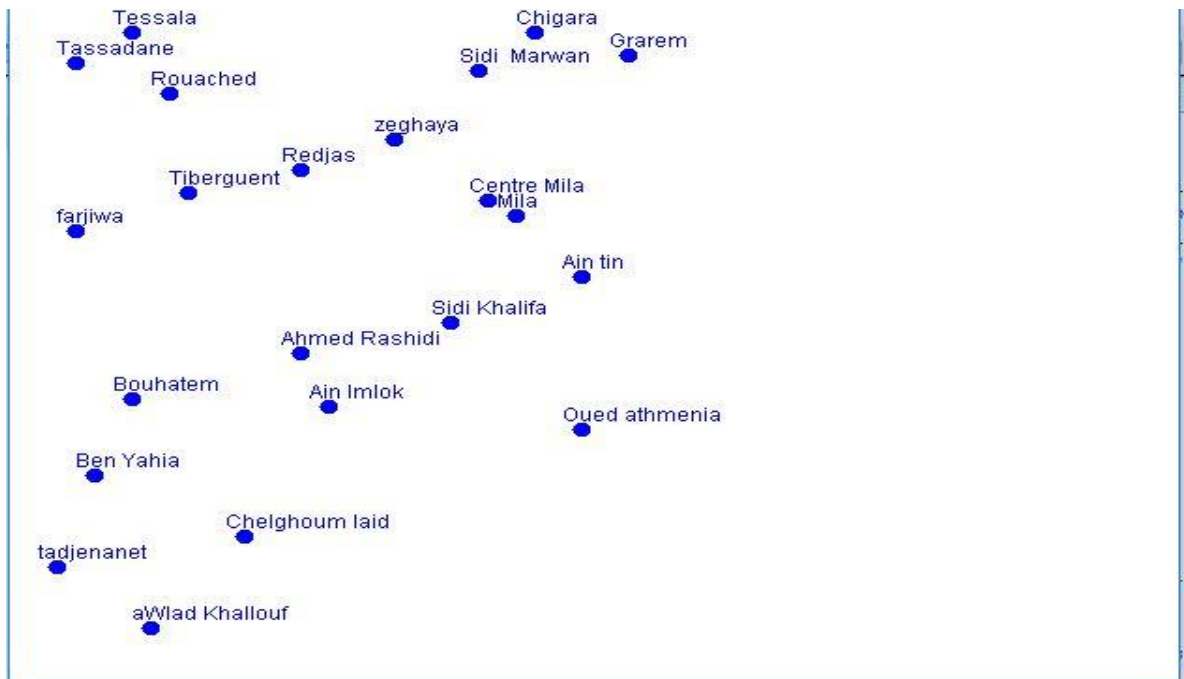


Figure 4.23 Carte des villes de Mila

Système d'ordonnancement et Planification des intervention des équipes de SONELGAZ

**Liste des Ville**

Numero	Nom
1	Mila
2	Oued athmenia
3	Chelghoum laid
4	farjiwa
5	zaghaya
6	Redjas
7	tadjenanet
8	Ahmed Rashidi
9	aWlad Khallouf
10	Ben Yahia
11	Bouhatem

**Ajouter**

**Liste de demande**

Num	Num Ville	Nom Ville
1	25	Oued Segan
2	4	farjiwa
3	1	Mila
4	4	farjiwa
5	16	Sidi Khalifa
6	21	Grarem
7	26	Tleghma
8	3	Chelghoum...
9	7	tadjenanet
10	22	ain bida ahr...
11	26	Tleghma

Population:

Generation:

% Croisement:

% Mutation:

Limite:

**Genérer**

**Reintialiser**

Figure4.24: remplir des informations

génération numéro 30  
fitness = 1246.0

#Demande	#Ville	Ville
0	0	Centre Mila
8	3	Chelghoum laid
10	22	ain bida ahrich
26	4	farjiwa
9	7	tadjenanet
24	17	Sidi Marwan
0	0	Centre Mila
5	16	Sidi Khalifa
18	2	Oued athmenia
3	1	Mila
14	3	Chelghoum laid

Meilleure tournée génération numéro 5  
Meilleure fitness = 1075.0

#Demande	#Ville	Ville
0	21	Grarem
17	24	beni ghacha
10	22	ain bida ahrich
2	4	farjiwa
0	0	Centre Mila
8	3	Chelghoum laid
22	23	mchira
25	10	Ben Yahia
20	21	Grarem
1	25	Oued Segan
0	0	Centre Mila

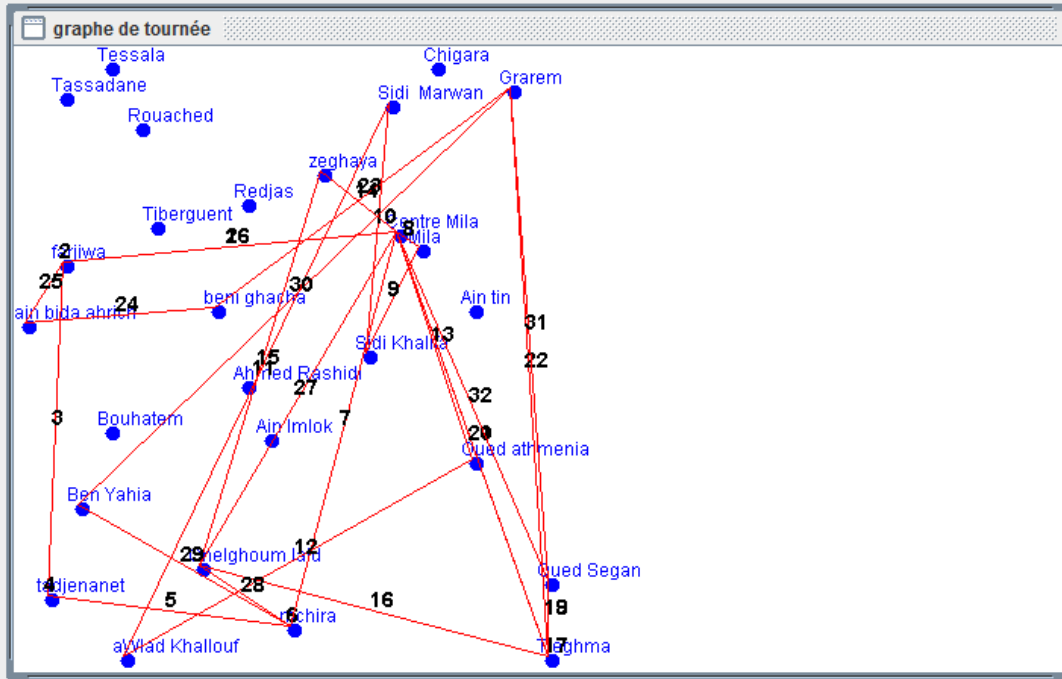


Figure 4.25 : exemple d'une solution du problème.

## 4.7 Testes et Comparaison

Après plusieurs visites et négociation avec les représentants de l'agence de SONELGAZ au niveau de la wilaya de Mila, nous avons réussi à accumuler des données des groupes de l'agence pendant une semaine de travail, ce qui nous permet de bien mesurer l'efficacité de notre méthode par rapport à la réalité appliquée, la figure suivante montre la liste des demandes des citoyens de la wilaya de Mila pendant une semaine :





LISTE DES AFFAIRES A ETUDIER

N° Affaire	Nature	Intitule	Adresse	Client
251501763	G01	AL EN GAZ NAT CITE GUENIFA YOUCEF OUED	CITE GUENIFA OUED SEGUIN Oued Seguin	APC OUED SEGUIN
251501764	G01	AL EN GAZ NAT KHARIFA MED MECHETA	MECHETA TIRKEF FERDJIOUA Ferdjioua	APC FERDJIOUA
251501765	G01	AL EN GAZ NAT BOUABDELLAH TAYEB ET GPES	MILA Mila	APC MILA
251501799	G01	AL EN GAZ NAT 20 LOGTS + LOCAUX + 05 LOGTS	FERDJIOUA Ferdjioua	ETP BOUGUessa ET HAD MESSAOUD
251501800	G01	AL EN GAZ NAT BENCHEIKH EL HOCINE AYAD	CITE ZAOUIA SIDI KHELIFA Sidi Khelifa	BENCHEIKH EL HOCINE AYAD
251501801	G01	AL EN GAZ DEFFOUS SALAH RUE BAGHRICHE	MTA EL HADRIA Grarem Gouga	DEFFOUS SALAH
251501802	G01	AL EN GAZ NAT MENACER KAMEL ET GPES	CITE MOSQUEE EL ATIQUE TELEGHM Téléghma	MENACER KAMEL
251501803	G01	AL EN GAZ NAT LOGT BOULTIF DOUADI CITE	CITE AFFES CH.LAID Chelghoum Laid	BOULTIF DOUADI
251501804	G01	AL EN GAZ NAT TAIBI FARID OULED KHLOUF	OULED KHLOUF TADJENANET Tadjenanet	TAIBI FARID
251501805	G01	AL EN GAZ NAT RAS EL AIN NADJI MECHETA	MECHETA REZAINE A.B.AHRICHE Ain Beida Ahriché	RAS EL AIN NADJI
251501806	G01	AL EN GAZ NAT NEBACHE ABDELMADJID ET	OULED SMAIL TELEGHMA Téléghma	NEBBACHE ABDELMADJID
251501807	G01	AL EN GAZ NAT SIGUAA ABDELMALEK ET GPES	LOTS 241 LOTS M'CHIRA M'Chira	SIGUAA ABDELMALEK
251501808	G01	AL EN GAZ NAT TAIBI SALIH ET GPES OULED	OULED KHLOUF Ouled Khelouf	TAIBI SALIH
251501809	G01	AL EN GAZ NAT OUCHERIF ALI ZAC CH.LAID	ZAC CH.LAID Chelghoum Laid	OUCHERIF ALI
251501810	G01	AL EN GAZ NAT BELLOUT ABDELHAFID	MECHTA DEKHLET BELOUM TELEGHMA Téléghma	BELLOUT ABDELHAFID
251501811	G01	AL EN GAZ NAT KHERKHARI ABDELHAFID	MECHETA DEKHLET BELLOUM TELEGHM Téléghma	KHERKHARI ABDELHAFID
251501812	G01	AL EN GAZ NAT KADRI EL YAZID Y.B.GUECHA	Y.B.GUECHA MERKEZ Yahia Béni guecha	KADRI EL YAZID

LISTE DES AFFAIRES A ETUDIER

N° Affaire	Nature	Intitule	Adresse	Client
251501813	G01	AL EN GAZ NAT ZOUAOUI TAYEB DJEBEL	DJEBEL AUGUEB OUED ATHMANIA Oued Athmania	ZOUAOUI TAYEB
251501814	G01	AL EN GAZ NAT MECHETA OULED GHROUSSE	OUED SEGUIN Oued Seguin	APC O.SEGUIN
251501816	G01	AL EN GAZ NAT HAIOUR AMOR MECHETA	MECHETA CHERAROU GRAREM Grarem Gouga	HAIOUR AMOR
251501817	G01	AL GN G.H.I MECHTA KEF+MESDOURA	CH-LADI M'Chira	APC CH-LAID
251501818	G01	AL GN G.H.I MECHTA MEZDOURA	CH-LADI M'Chira	APC CH-LAID
251501830	G01	AL EN GAZ NAT BOUDECHICHE YASSINE	MECHETA BOUFOUH ZEGHAIA Zéraia	BOUDECHICHE YASSINE
251501831	G01	AL EN GAZ NAT CHERBIT TAYEB MECHETA	CITE ZAOUIA SIDI MEROUANE Sidi Merouane	CHERBIT TAYEB
251501832	G01	AL EN GAZ NAT MECHETA TIHELOUFIN	APC BENYAHIA ABDERAHMANE Ben Yahia	APC BENYAHIA
251501833	G01	AL EN GAZ NAT BOUCHENINE NAAMOUNE ET	MECHETA AIN EL HAMRA FERDJIOUA Ferdjioua	BOUCHENINE NAAMOUNE
251501835	G04	AUGMENTATION DU DEBIT PICINE SEMI	TADJENANET Tadjenanet	DJS MILA

Le Chef DRC

ج. حريزي

Figure 4.26 : requêtes de citoyens au service technique du SONEELGAZ (Données réels d'une semaine)

D'après les responsables du service technique de la société, les groupes travaillent 8 heures par jour, et en consultant le registre du suivi des taches, on a constaté que chaque travail prend un moyen d'une heure, la figure suivante montre un capture de suivi de quelques travaux.

15.04.17	SMARA	10h42 11h40 11h05 11h06 11h48 11h50	DT Dept mechtla Smara au C.M.056 R ⊖ IV J 5282. F dept Smara ⊕ F J 5282 ⊖ avec IV Immediate F I 223 prise de secours sur Dept Telghme.	S
11.04.17	B.CH.	1300 13:40 13:45 13:47	DT I 238 IV TRN:610 entre pote n° 1240 et nouveau pote 241 lots. R ⊕ I 238. Fermeture boucle au p N° 1098	S
12.04.17	BCH.	18h00	I boucle au pote n° 1098	
02.04.17	ZAC	10h00 11h00	Fer le J 5556 IV Cellule Protection P 1119 Fer Cellule Protection	S
02.04.17	IT Tadj	10h40 11h00	IV Dept P 11663 au P 1136	S

Figure 4.27 Exemple de déroulement d'un travail

Pour bien mesurer le temps total d'un travail y compris le temps de déplacement, on a fixé la vitesse moyenne des véhicules à 60 km/h. Notre algorithme génétique se base dans ses calculs sur la distance, pour cela, on a transformé les 8 heures de travail des groupes vers une distance en utilisant la loi suivante :

60 Km  $\longrightarrow$  1h  
 Totale  $\longrightarrow$  8h

La distance totale parcourue durant les 8 heures est 480 Km qui représentent la durée maximale d'une tournée.

La société Sonelgaz traite les demandes par ordre chronologique de leur temps d'arrivée, ce qui nous donne le résultat suivant qui est réalisé dans 7 jour :

### 1<sup>er</sup> jour :

Centre Mila  $\longrightarrow$  Oued segun  $\longrightarrow$  ferdjioua  $\longrightarrow$  Mila  $\longrightarrow$  Ferdjioua  $\longrightarrow$  Centre Mila

### 2eme Jour :

Centre Mila  $\longrightarrow$  Sidi khelifa  $\longrightarrow$  Grarem  $\longrightarrow$  Teleghma  $\longrightarrow$  Chelghoum  $\longrightarrow$  Centre Mila

### 3eme Jour :

Centre Mila → Tadjanet → Ain Baida ahrich → Teleghma → Centre Mila

**4eme jour :**

Centre Mila → Mchira → Ouled Khlouf → Chelghoum → Teleghma → Teleghma  
 Centre Mila ←

**5eme Jour :**

Centre Mila → Bni guecha → Oued Athmania → Oued Segun → Grarem → Centre

**6eme Jour :**

Centre Mila → Mchira → Mchira → Zeghaia → Sidi Merouane → Centre Mila

**7eme Jour :**

Centre Mila → Ben Yahia → Ferdjioua → Tadjanet → Centre Mila.

La figure suivante représente cette solution appliquée dans la réalité sous forme graphique.

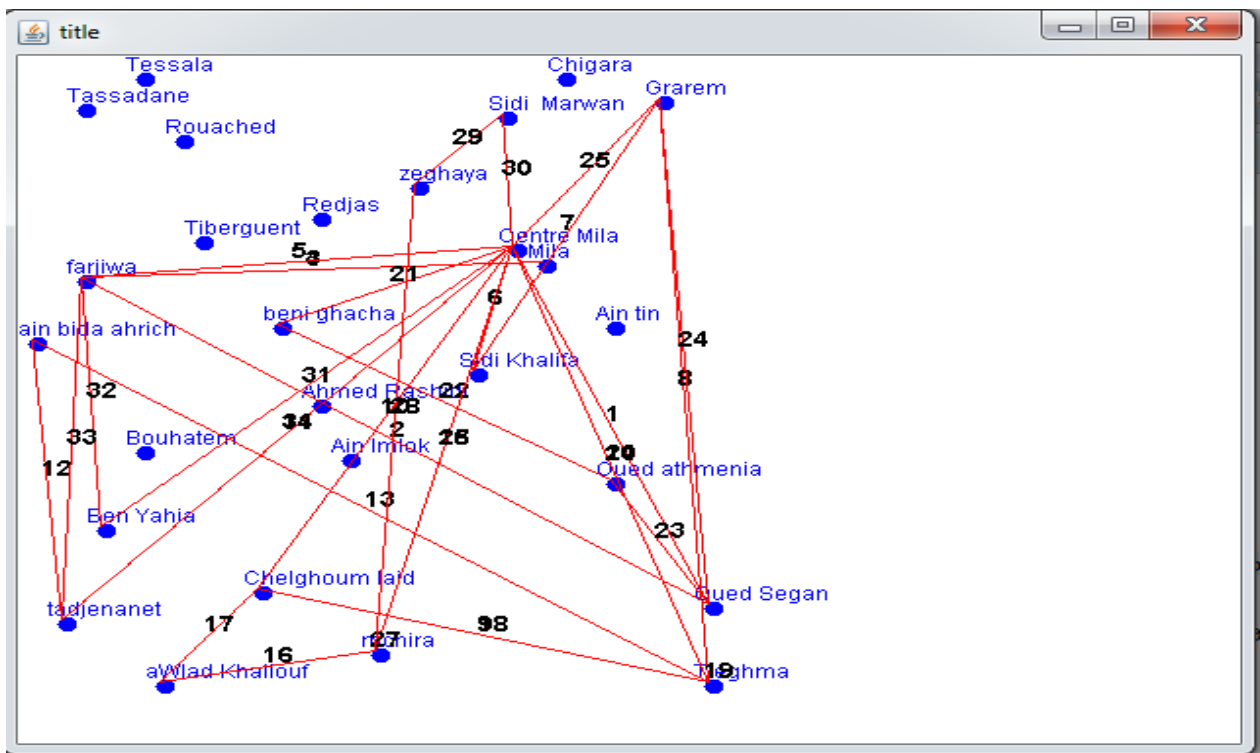


Figure 4.28 Les tournées appliquées par le groupe technique de la société.

On a utilisé les même données réelles pour tester l'algorithme génétique développé, la première étape c'est le saisie de ces données, la figure suivante montre cette étape.



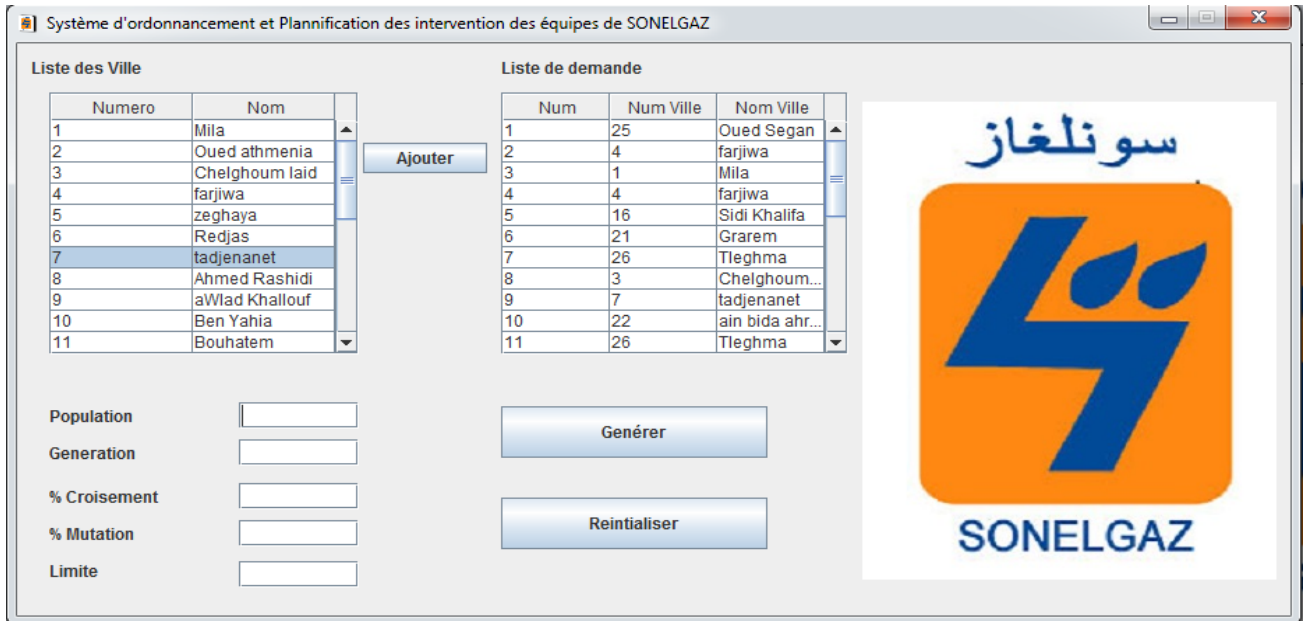


Figure 4.29 : L'ajout des demandes

La deuxième étape consiste à fixer les paramètres de l'algorithme génétique qui sont la taille de la population, le nombre des itérations, la probabilité de croisement et la probabilité de mutation, et la taille maximale d'une tournée. La figure suivante montre les valeurs de ces paramètres.

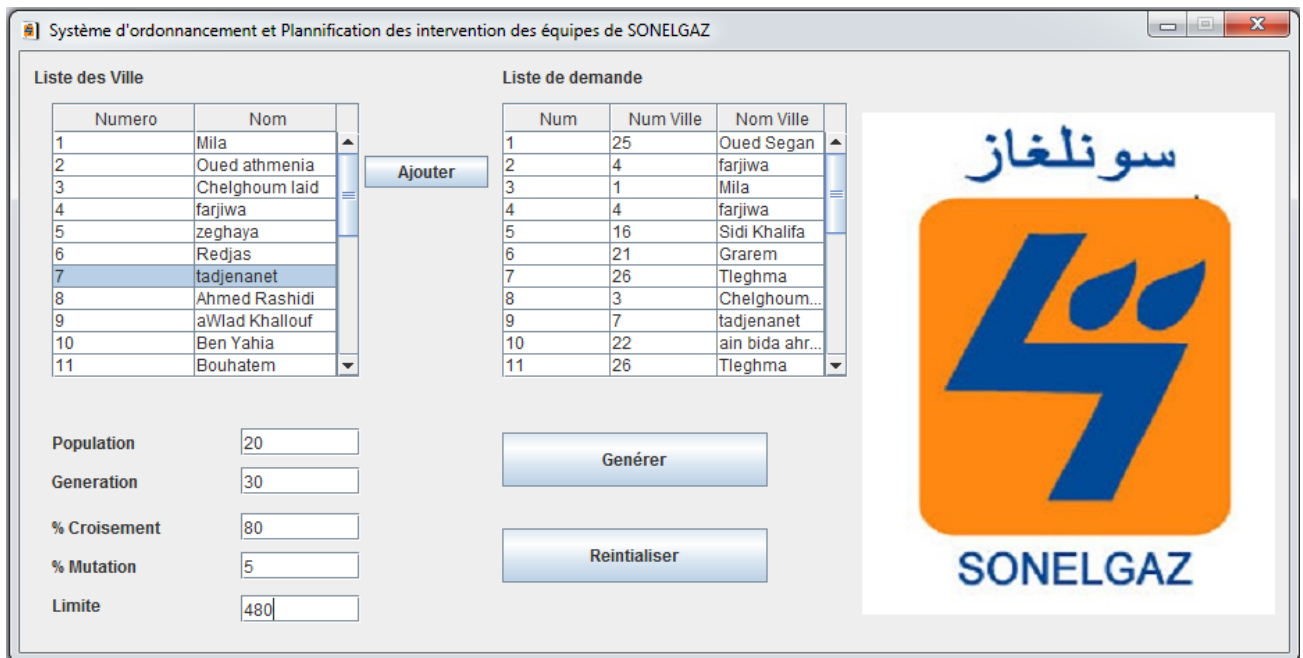


Figure 4.30 Saisir les informations

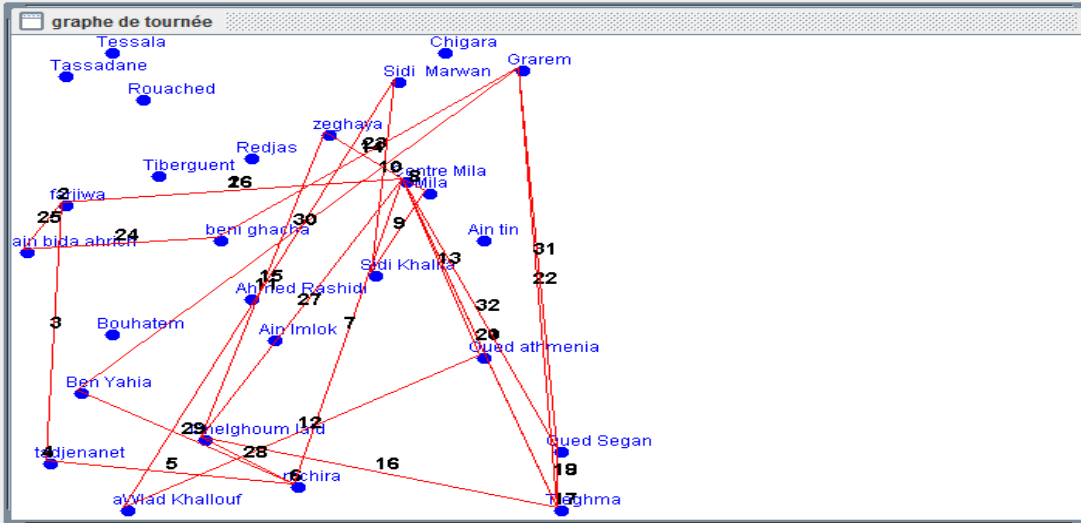
En cliquant sur le bouton Générer, on obtient le résultat suivant :

génération numéro 30  
fitness = 1246.0

#Demande	#Ville	Ville
0	0	Centre Mila
8	3	Chelghoum laid
10	22	ain bida ahrich
26	4	farjiwa
9	7	tadjenanet
24	17	Sidi Marwan
0	0	Centre Mila
5	16	Sidi Khalifa
18	2	Oued athmenia
3	1	Mila
14	2	Chelghoum laid

Meilleure tournée génération numéro 5  
Meilleure fitness = 1075.0

#Demande	#Ville	Ville
0	21	Grarem
17	24	beni ghacha
10	22	ain bida ahrich
2	4	farjiwa
0	0	Centre Mila
8	3	Chelghoum laid
22	23	mchira
25	10	Ben Yahia
20	21	Grarem
1	25	Oued Segan
0	0	Centre Mila



Meilleure tournée génération numéro 5  
Meilleure fitness = 1075.0

#Demande	#Ville	Ville
0	0	Centre Mila
26	4	farjiwa
4	4	farjiwa
27	7	tadjenanet
9	7	tadjenanet
12	23	mchira
21	23	mchira
0	0	Centre Mila
3	1	Mila
5	16	Sidi Khalifa
24	17	Sidi Marwan

Meilleure tournée génération numéro 5  
Meilleure fitness = 1075.0

#Demande	#Ville	Ville
0	0	Centre Mila
3	1	Mila
5	16	Sidi Khalifa
24	17	Sidi Marwan
13	9	ayWlad Khallouf
18	2	Oued athmenia
0	0	Centre Mila
23	5	zaghaya
14	3	Chelghoum laid
7	26	Tleghma
16	26	Tleghma

Meilleure tournée génération numéro 5  
Meilleure fitness = 1075.0

#Demande	#Ville	Ville
7	26	Tleghma
16	26	Tleghma
19	25	Oued Segan
11	26	Tleghma
0	0	Centre Mila
15	26	Tleghma
6	21	Grarem
17	24	beni ghacha
10	22	ain bida ahrich
2	4	farjiwa
0	0	Centre Mila

Meilleure tournée génération numéro 5  
Meilleure fitness = 1075.0

#Demande	#Ville	Ville
0	21	Grarem
17	24	beni ghacha
10	22	ain bida ahrich
2	4	farjiwa
0	0	Centre Mila
8	3	Chelghoum laid
22	23	mchira
25	10	Ben Yahia
20	21	Grarem
1	25	Oued Segan
0	0	Centre Mila

Figure 4.31 les résultats de notre application.

## 4.8 Analyse des résultats

Le résultat qui représente la réalité appliquée dans la société se compose de 7 tournées qui prennent 7 jours pour l'accomplir si on a un seul groupe technique.

La solution qu'on a obtenue par application de l'algorithme génétique se compose de 5 tournées seulement, c'est-à-dire qu'on peut la réaliser en 5 jours par un groupe technique. Donc l'algorithme génétique nous permet de gagner deux jours de travail d'une équipe ce qui montre son efficacité.

L'algorithme génétique peut nous donner des résultats meilleurs que celui qu'on a obtenu par plusieurs tests en changeant les paramètres de l'algorithme génétique.

## **4.9 Conclusion**

Dans ce chapitre nous avons présenté l'application développée, et les outils informatique utilisés, puis on a expliqué les données réelles apporté de l'agence de SONELGAZ au niveau de la wilaya de Mila, enfin on a terminé par une comparaison de la solution appliquée dans la réalité avec laquelle obtenu par l'algorithme génétique développé. Cette comparaison montre l'efficacité de de l'application des méthodes d'optimisation sur ce type de problèmes.

## **Conclusion générale**

Les problèmes de transport font partie intégrante de la vie quotidienne des décideurs et des planificateurs. Il s'agit de déterminer les tournées d'une flotte de véhicules afin de livrer une liste de clients, ou de réaliser des tournées d'interventions (maintenance, réparation, contrôles) ou de visites (visites médicales, commerciales, etc.). Le but est de minimiser le coût de livraison des biens.

Dans le cadre de ce mémoire, nous nous sommes intéressés à une importante variante de ces problèmes qui est le problème de transport à la demande. Ce problème, consiste à déterminer les tournées et les horaires pour les véhicules qui effectuent le transport d'usagers à leur demande, d'une origine à une destination en minimisant le coût total des tournées.

En fait, on a traité un problème très fréquent dans notre vie, c'est le problème des interventions des équipes de maintenance de la société de SONELGAZ, ces interventions sont arrivées dans la plus part du temps après des jours et des fois des semaines du lancement de la requête, ce qui pose les citoyens dans des situations embarrassantes, ou même dangereuses.

Pour aider les décideurs de cette société à bien utiliser les ressources humaines et matérielles du service technique.

On a traité ce problème comme étant un problème de transport à la demande, le transport représente le déplacement des équipes techniques de la société, selon les requêtes des citoyens. On a développé un outil informatique qui utilise la méthode des algorithmes génétique pour résoudre ce problème.

Les résultats obtenus montrent l'efficacité de cette méthode par rapport à la solution appliquée sur terrain, encourage les étudiants à utiliser d'autres méthodes dans le but d'améliorer les résultats et ouvre la porte pour traiter d'autres problèmes de la vie quotidien par des méthodes similaires.

## **Références**

- [1] Baptiste Mille, « Méthodes approchées pour la résolution d'un problème d'ordonnancement avec travaux interférents », mémoire fin d'étude, École Polytechnique de l'Université de Tours.
- [2] Sidi Mohamed Douiri et autre, «Cours des Méthodes de Résolution Exactes Heuristiques et Métaheuristiques », Université Mohammed V RABAT.
- [3] Hacene halim – Mémoire fin d'étude.
- [4] Joseph Ayas « La Recherche Tabou », Université technique de la catalonie Espagne.
- [5] Mr A.Bourideh – « Cours Optim-Combinatoire » Centre Univ-de-Mila 2016.
- [6] AYOUB CHENOUKI « Approche génétique pour le problème de PVC » mémoire Master Université Constatine ,2016.
- [7] S. Ben Ismail «Introduction à l'optimisation combinatoire», cours pédagogique de l'université Telecom Bretagne.
- [8] Christelle Reynès « Etude des Algorithmes génétiques et application aux données de protéomique » Thèse de doctorat Université de MONTPELLIER I.
- [9] ZIDI ISSAM « Modélisation et Optimisation d'un Système de Transport à la Demande Multicritère et Dynamique »Thèse de doctorat de l'école centrale de Lille ,2012.
- [10] Michel Nabaa « Approche décentralisée pour résoudre le problème du transport à la demande » Thèse de doctorat de l'université de France, 2009.
- [11] Jamal Rebaine « Technique de branch and bound », cours hiver département d'informatique et mathématique Université du Quebec, 2005.
- [12] M. Dorigo et autres « Ant colony optimisation » IEEE computational intelligence magazine volume 1, 2006.