



UNIVERSITE
Abdelhamid Ibn Badis
MOSTAGANEM

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM

Faculté des Sciences Exactes et de l'Informatique
Département de Mathématiques et d'Informatique
Filière : Informatique

MEMOIRE DE FIN D'ETUDES
Pour l'Obtention du Diplôme de Master en Informatique
Option : **Ingénierie des Systèmes d'Information**

THEME :
Applications des algorithmes génétiques

Présenté par:

- BELHIA SLIMANE
- BETTAHER MOHAMMED

Encadré par :

- Mme. AMINA DELALI

Année Universitaire 2018/2019

Dédicaces

Avant tout, je dois rendre grâce à dieu de m'avoir donné le courage de terminer
ce travail.

Je dédie ce travail à ma famille, ma mère, mon père, ma chère femme, mon fils
iyado, mes frères, mes sœurs, ainsi mon binome et tous mes amis de promotion
2^{ème} année Master Informatique.

Et toutes les personnes qui nous ont aidés de réaliser ce travail.

BELHIA Slimane

Dédicaces

Avant tout, je dois rendre grâce à dieu de m' avoir donné le courage de terminer
ce travail.

Je dédie ce travail à ma mère, ainsi mon binome et tous mes amis de promotion
2^{ème} année Master Informatique.

Et toutes les personnes qui nous ont aidés de réaliser ce travail.

BETTAHER Mohammed

Remerciements

Avant de présenter ce travail, nous tenons à remercier Dieu tout puissant, de nous avoir permis d'arriver à ce niveau d'étude.

Nous tiendrons à remercier toutes les personnes qui ont contribué au succès de notre travail.

Tout d'abord, nous adressons nos remerciements à notre encadreur, Madame « **DELALI AMINA** » qui nous a beaucoup aidé dans notre travail.

Enfin, nous tiendrons à remercier toutes les personnes qui nous ont conseillé et relu lors de la rédaction de ce travail : nos familles , nos amis et camarades de promotions.

Liste des figures

Figure1 : diagramme d'activité de l'algorithme génétique	6
Figure2 : Solution S d'un VRP	18
Figure3 : Solution S annotée.....	18
Figure4 : voyageur de commerce (la carte du France)	21
Figure 5 : Interface de StarUML version 3.1.0	26
Figure 6 : Diagramme de cas d'utilisation	27
Figure7 : Diagramme d'activité.....	28
Figure 8 : Diagramme de classe.....	29
Figure 9 : l'interface graphique principale pour importer un fichier .csv	32
Figure 10 : l'interface graphique de fichier csv dans sa forme originale.....	32
Figure 11 : Interface graphique de paramètres des algorithmes génétiques.....	33
Figure 12 : Interface graphique de chemin final	33

Liste des tableaux

Tableau 1 : Sélection par rang pour un problème de maximisation	7
Tableau 2 : Sélection par la roulette pour un problème de maximisation.....	8
Tableau 3 : L'opérateur de croisement à 1 point.....	9
Tableau 4 : L'opérateur de croisement à 2 point.....	9
Tableau 5 : Opérateur de mutation d'un bit.....	10
Tableau 6 : Tableau de Complexité du problème	22
Tableau 7: Tableau de Statistique sur les Paramètres de l'AG	22

Liste des abréviations

Abréviation	Expression Complète	Page
AG	Algorithme génétique	1
PX	Probabilité de croisement	4
PM	probabilité de mutation	5
VRP	Vehicle Routing Problème	17
OX	Opérateur ordre crossover	19
PVC	Problème du voyageur de commerce	22

Table des matières

Introduction générale	1
Chapitre 01: Algorithmes génétiques concepts et théorie.....	2
1.1 Introduction.....	3
1.2 Les Algorithmes génétiques	3
1.3 Concepts de base	4
1.4 Principes de base des AG	4
1.5 Fonctionnement des AG	5
1.5.1 Codage du chromosome	5
1.5.2 Génération de la population initiale	6
1.5.3 Méthodes de sélection	6
1.5.4 Opérateurs de croisement	9
1.5.5 Opérateur de mutation	10
1.5.6 Méthode d'insertion.....	10
1.5.7 Test d'arrêt :.....	11
1.6 Conclusion :	12
Chapitre 02: Problèmes réels par domaine.....	13
2.1 Introduction	14
2.2 Domaine de transport aérien : Gestion du roulage sur les grands aéroports	14
2.2.1 Modélisation	14
2.2.2 AG : algorithmes génétiques	15
2.2.2.1 Codage des données	15
2.2.2.2 Fonction d'adaptation	15
2.2.2.3 Croisement et Mutation	16
2.2.2.4 Sharing	16
2.2.2.5 Critère d'arrêt.....	17
2.2.2.6 Résolutions par Clusters	17
2.3 Les problèmes de tournées de véhicules	17
2.3.1 L'algorithme génétique interactif pour le VRP	17
2.4 Conclusion	19
Chapitre 03: Problème du voyageur de commerce	20
3.1 Introduction.....	21
3.2 Théoriquement	21
3.2.1 Pratiquement.....	22
3.2.2 Complexité du Problème	22
3.2.3 Intérêt.....	22
3.3 Application de l'AG au PVC.....	23
3.3.1 L'Objectif	23
3.3.2 Les Statistiques sur les Paramètres de L'AG	23
3.4 Conclusion	24

Chapitre 04: Conception et Implémentation	25
4.1 Introduction.....	26
4.2 Conception.....	26
4.2.1 Diagramme de cas d'utilisation	27
4.2.2 Diagramme d'activité	28
4.2.3 Diagramme de classe	29
4.3 Implémentation	30
4.3.1 Les coordonnées géographiques	30
4.3.1.1 Définition de la latitude	30
4.3.1.2 Définition de la longitude	30
4.3.1.3 Les différents référentiels de coordonnées Géographiques :	30
4.3.2 Environnement matériel	31
4.3.3 Environnement logiciel.....	31
4.3.4 Langage et outils de développement	31
4.4 Les interfaces graphiques principales.....	31
4.5 Conclusion	34
Conclusion générale.....	36
Bibliographie.....	37
Webographie	38

Résumé

Les algorithmes génétiques sont des algorithmes d'optimisation stochastique fondés sur les mécanismes de la sélection naturelle et de la génétique. Leur fonctionnement est destiné à des problèmes complexes. Ils peuvent jouer un rôle intéressant dans le cadre de la protéinique.

La première partie est consacrée aux algorithmes génétiques, concepts et théorie, ainsi que des variations existantes. La partie suivante traite des problèmes réels et solutions correspondantes. La dernière partie aborde une application particulière au problème du voyageur de commerce et une description de la problématique.

Mots-clés : algorithmes génétiques, le problème du voyageur de commerce.

Introduction générale

Introduction générale

Parmi tous les types d'algorithmes existants, certains ont la particularité de s'inspirer de l'évolution des espèces dans leur cadre naturel. Ce sont les algorithmes génétiques. Les espèces s'adaptent à leur cadre de vie qui peut évoluer, les individus de chaque espèce se reproduisent, créant ainsi de nouveaux individus, certains subissent des modifications de leur ADN, certains disparaissent etc.

Les algorithmes génétiques sont des algorithmes évolutionnistes d'optimisation stochastique inspirés des mécanismes de l'évolution naturelle (sélection, adaptation, reproduction, recombinaison, mutation) élaborés par Charles Darwin. Ils sont des algorithmes itératifs de recherche globale dont le but est d'optimiser une fonction prédéfinie appelée critère ou fonction coût "fitness". Ils permettent de faire évoluer un ensemble initial de solutions vers un ensemble final. Les algorithmes génétiques sont utilisés dans l'automatique, le but de cette utilisation est d'optimiser au maximum les gains des régulateurs utilisés dans la commande. A partir des années 1990, quelques publications évoquent l'utilisation des AG pour la résolution de problèmes d'optimisation dans le domaine de la commande de systèmes informatiques.

Dans notre projet nous allons définir les algorithmes génétiques et leur fonctionnement, ainsi que les concepts et les principes de base.

L'objectif de notre travail est d'appliquer l'algorithme génétique au problème du voyageur du commerce. Mais nous donnerons d'autres exemples d'application.

Le reste du rapport est organisé en quatre chapitres suivis d'une conclusion générale.

Chacun représente une partie du travail de ce projet qu'elle soit théorique ou pratique :

- Chapitre 1 : Algorithmes génétiques concepts et théorie
- Chapitre 2 : Problèmes réels par domaine
- Chapitre 3 : Problème du voyageur de commerce
- Chapitre 4 : Conception et réalisation

Chapitre 01

Algorithmes génétiques concepts et théorie

1.1 Introduction

Nous avons assisté ces dernières années à une croissance très rapide des travaux utilisant les algorithmes génétiques (AG). Nous présentons dans ce chapitre la définition des algorithmes génétiques, les principes et les fonctionnalités de base d'un algorithme génétique, les différentes opérations génétiques telles que la mutation et le croisement.

1.2 Les Algorithmes génétiques

Les AGs ont été initialement développés par John Holland (1975)¹. C'est au livre de Goldberg (1989)² que nous devons leur popularisation. Leurs champs d'application sont très vastes. Outre l'économie (minimisation du risque des portefeuilles), ils sont utilisés pour l'optimisation de fonctions, en finance, en théorie du contrôle optimal (recherche opérationnelle), ou encore en théorie des jeux répétés et différentiels (en l'occurrence dans les jeux évolutionnaires et le dilemme du prisonnier) et la recherche d'information (Google) ainsi que la recherche des plus courts chemins en théorie des graphes (routages Internet ou GPS). La raison de ce grand nombre d'applications est claire : simplicité et efficacité. Bien sûr, d'autres techniques d'exploration stochastiques existent, la méthode de Monte-Carlo peut être considérée comme un concept similaire.

Pour résumer, Lerman et Ngouet (1995)³ distinguent quatre principales propriétés qui font la différence fondamentale entre ces algorithmes et les autres méthodes :

P1. Les algorithmes génétiques utilisent un codage des paramètres, et non les paramètres eux-mêmes.

P2. Les algorithmes génétiques travaillent sur une population de points, au lieu d'un point unique.

P3. Les algorithmes génétiques n'utilisent que les valeurs de la fonction étudiée, pas sa

¹ Holland, J. H. "Adaptation in Natural and Artificial Systems". Ann Arbor : University of Michigan Press. 1975. Libraries Australia.

² Goldberg, D. "GeneticAlgorithm In Search, Optimization And Machine Learning". Addison-Wesley Longman. 1989. Boston, USA.

³ Lerman, I. &Ngouenet, F. "Algorithmes génétiques séquentiels et parallèles pour une représentation affine des proximités".Rapport de Recherche de l'INRIA.1995. Rennes France.

dérivée, ou une autre connaissance auxiliaire.

P4. Les algorithmes utilisent des règles de transition probabilistes, et non déterministes.

La simplicité de leurs mécanismes, la facilité de leur mise en application et leur efficacité même pour des problèmes complexes ont conduit à un nombre croissants de travaux ces dernières années.

1.3 Concepts de base

La transposition des concepts biologiques dans un cadre artificiel a conduit à la définition des concepts suivants :

- Individu : structure fondamentale permettant d'encoder une solution candidate au problème posé.
- Population : ensemble d'individus d'une même génération.
- Fonction d'adaptation : mesure d'efficacité des individus solutions, régissant les transformations génétiques appliquées.
- Opérateurs génétiques : procédures de transformation des individus entre deux générations.

1.4 Principes de base des AG

Indépendamment de problématique traitée, les algorithmes génétique sont basés sur six principes :

1. Choisir le codage des solutions.
2. Générer une population initiale de taille fixe N , formée d'un ensemble fini de solutions, dit génération initiale.
3. Définir une fonction par un mécanisme de sélection qui choisit pour un éventuel couplage.
4. Générer de nouvelles solutions à l'aide des opérateurs génétiques en utilisant :
 - Opérateur de croisement : il manipule la structure des chromosomes des parents afin de produire des individus meilleurs ou différents. Cet opérateur est effectué selon une probabilité P_x .

- Opérateur de mutation : il évite d'établir de populations uniformes incapables d'évaluer. Il consiste à modifier des valeurs des gènes de chromosomes selon une probabilité de mutation P_m .
5. Etablir un compromis entre les solutions produites (progénitures) et les solutions productrices (les parents) en utilisant un mécanisme d'insertion. En d'autres termes, et suite à des informations précises, décider ce qui doit rester et ce qui doit disparaître. Tout ceci, en sauvegardant à chaque génération une taille de la population N fixe.

1.5 Fonctionnement des AG

L'algorithme génétique, présente dans la figure (Figure 1), débute par une génération d'une population initiale de N individus, pour lesquels, nous calculons les valeurs de leur fonction objective et nous sélectionnons les individus par une méthode de sélection. Les individus, sujets de croisement par l'opérateur de croisement, sont choisis selon une probabilité P_x . Leurs résultats peuvent être mutés par un opérateur de mutation avec une probabilité de mutation P_m . Les individus issus de ces opérateurs génétiques seront insérés par une méthode d'insertion dans la nouvelle population dont nous évaluons la valeur de la fonction objective de chacun de ses individus. Ainsi, ce test est vérifié alors l'algorithme s'arrête avec une solution optimale, test d'arrêt sera effectué pour vérifier la qualité des individus obtenus. sinon on réitérera le processus pour la nouvelle génération.

1.5.1 Codage du chromosome

Le choix du codage des données dépend de la spécificité du problème traité. Il conditionne fortement l'efficacité de l'algorithme génétique. Un chromosome (une solution particulière) a différentes manières d'être codé selon l'alphabet. Nous distinguons trois types de codages :

- Numérique si l'alphabet est constitué de chiffres.
- Symbolique si l'alphabet est un ensemble de lettres alphabétiques ou des symboles.
- Alphanumérique si nous utilisons un alphabet combinant les lettres et les chiffres.

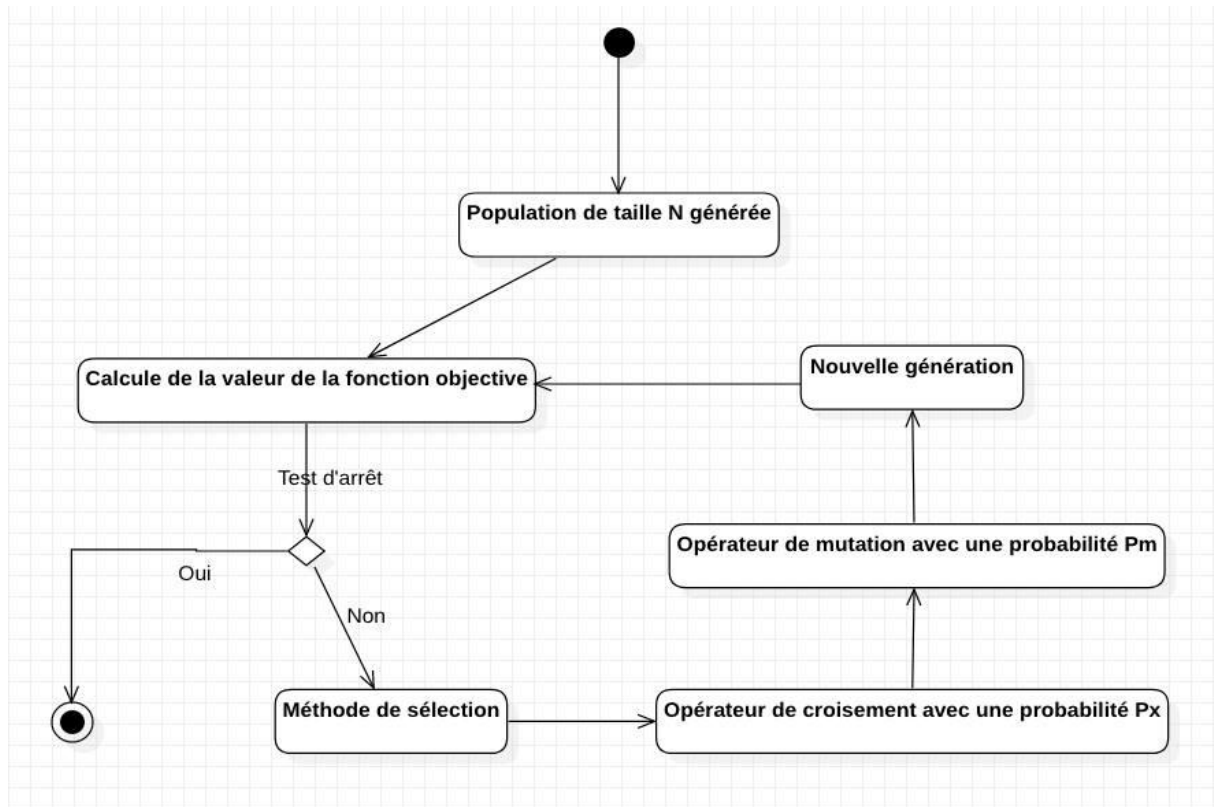


Figure1 : diagramme d'activité de l'algorithme génétique

1.5.2 Génération de la population initiale

Dans les problèmes d'optimisation, une connaissance de « candidats de bonne qualité » comme ponts d'initialisation conditionné la rapidité de la convergence vers l'optimum. Si la position de l'optimum dans l'ensemble des réalisables est totalement inconnue, il est naturel de générer aléatoirement des individus. Les tirages sont réalisés en respectant les contraintes et d'une manière uniforme dans chacun des domaines associés aux composantes de l'ensemble de solutions. Des connaissances à priori sur le problème traité permettent de générer les individus dans un domaine particulier afin d'accélérer la convergence de l'algorithme génétique. Les opérateurs de croisement et de mutation permettent d'entretenir la diversité d'une population non homogène au cours des générations afin de parcourir l'ensemble de solutions le plus largement possible.

1.5.3 Méthodes de sélection

La sélection permet d'identifier les individus susceptibles d'être croisés dans une population. Nous trouvons dans la littérature plusieurs principes de sélection :

Sélection par rang : Il consiste à attribuer à chaque individu son classement par ordre d'adaptation. Pour de maximisation, nous classons les individus selon l'ordre croissant des valeurs de la fonction objective. Ainsi, les plus mauvais individus (c'est-à-dire celui qui possède la plus petite valeur de la fonction objectif) prendra le numéro 1 et ainsi de suite (voir Tab 1). Pour un problème de minimisation, nous ordonnons les individus selon l'ordre décroissant. On prélève ensuite une nouvelle population à partir de cet ensemble d'individus ordonnés, en utilisant des probabilités indexées sur les rangs des individus :

$$\text{Probabilité de sélection (parent } i) = \frac{\text{Rang}(\text{Parent } i)}{\sum \text{Rang}(\text{Parent } j)}$$

Cette procédure est très simple et exagère le rôle du meilleur élément au détriment d'autres éléments potentiellement. La seconde, par exemple, aura une probabilité d'être sélectionné plus faible que le premier, bien qu'il soit peut-être situé dans une région d'intérêt.

Tableau 1 : Sélection par rang pour un problème de maximisation

	Chromosome	Fitness	Rang	Prob de sélection = $\frac{\text{Rang}}{\text{Total}(\text{Rang})}$
	Parent 1	30	2	33.33%
	Parent 2	60	3	50 %
	Parent 3	10	1	16.67%
Total		100	6	100%

Sélection par la roulette : Dans un problème d'optimisation de maximisation, on associe à chaque individu i une probabilité de sélection, noté Probi , proportionnelle à sa valeur F_i de la fonction objectif :

$$\text{Probi} = \frac{F_i}{\sum(F_j)}$$

Chaque individu est alors reproduit avec la probabilité Probi . Certains individus (les bons) seront alors « plus » reproduits et d'autres (les mauvais) éliminés (voir Tab 2)

Tableau 2 : Sélection par la roulette pour un problème de maximisation

	Chromosome	Fitness	Prob de sélection = $\frac{Fitness}{Total(Fitness)}$
	Parent 1	30	30%
	Parent 2	60	60 %
	Parent 3	10	10%
Total		100	100%

Pour un problème de minimisation, on utilise une probabilité de pour un individu i égal à : $\frac{(1-Prob_i)}{(N-1)}$

- **Sélection aléatoire** : La sélection se fait aléatoirement, uniformément et sans intervention de la valeur d'adaptation. Chaque individu a donc une probabilité uniforme ($1/N$) d'être sélectionné. En général, la convergence de l'algorithme génétique est lente en utilisant cette méthode.

- **Sélection par Tournoi** : Cette méthode de sélection augmente les chances des individus de mauvaise qualité par rapport à leur fitness, de participer à l'amélioration de la population. En effet, c'est une compétition entre les individus d'une sous-population de taille M ($M \leq N$) prise au hasard dans la population. Le paramètre M est fixé à priori par l'utilisateur. L'individu de meilleure qualité par rapport à la valeur de la fonction objective sera sélectionné pour l'application de l'opérateur de croisement. Le paramètre M joue un rôle important dans la méthode du tournoi.

Dans le cas où $M = N$ avec N est la taille de la population. Le résultat par la sélection de la méthode du tournoi donne à chaque fois un seul individu qui est le meilleur individu par rapport à la valeur de la fonction objective. Ce qui réduit l'algorithme génétique à un algorithme de recherche local travaillant sur une seule solution à la fois. Ce type d'algorithmes a pour inconvénient de converger parfois rapidement vers un optimum local.

Dans le cas $M = 1$, la méthode de sélection du tournoi correspond à la sélection aléatoire.

1.5.4 Opérateurs de croisement

Le croisement a pour but d'enrichir la diversité de la population en manipulant les composantes des chromosomes. Classiquement, les croisements sont envisagés avec deux parents et génèrent deux enfants. Il est appliqué avec une probabilité P_x , communément appelée probabilité de croisement. Après l'utilisation de la méthode de sélection pour le choix de deux individus, nous générons un nombre aléatoire $\alpha \in [0,1]$.

Si ($\alpha \leq P_x$), nous appliquons l'opérateur de croisement sur le couple. Les plus anciens opérateurs de croisement utilisés sont l'opérateur de croisement à un point et à deux points sur deux chromosomes à codage binaire. Ils constituent la base des opérateurs de croisement. L'opérateur à un point de croisement consiste à diviser chacun des deux parents en deux parties à la même position, choisie au hasard. L'enfant 1 est composé de la première partie parent et la deuxième partie du deuxième parent alors que l'enfant 2 est constitué de la

Première partie du deuxième parent et la deuxième partie du premier parent (TAB 3).

Tableau 3 : L'opérateur de croisement à 1 point

Parent 1	1	0	1	0	1	0	1	0	Enfant 1	1	0	1	0	0	1	0	0
Parent 2	0	0	1	0	0	1	0	0	Enfant 2	0	0	1	0	1	0	1	0

L'opérateur à deux points de croisement est illustré dans le tableau (TAB 4). Il consiste à fixer deux positions. L'enfant 1 sera la copie du parent 1 en remplaçant sa partie entre les deux positions par celle du parent 2. On effectuera la même opération pour déterminer l'enfant 2 en intervertissant les rôles des parent 1 et parent 2.

Tableau 4 : L'opérateur de croisement à 2 point

Parent 1	1	0	1	0	1	0	1	0	Enfant 1	1	0	1	0	0	1	1	0
Parent 2	0	0	1	0	0	1	0	0	Enfant 2	0	0	1	0	1	0	0	0

1.5.5 Opérateur de mutation

L'opérateur de mutation apporte aux algorithmes génétiques l'aléa nécessaire à une exploration efficace de l'espace. Cet opérateur nous garantit que l'algorithme génétique sera susceptible d'atteindre la plupart des points du domaine réalisable. Cerf en 1994⁴, a démontré théoriquement que l'algorithme génétique converge en probabilité en utilisant l'opérateur de mutation et sans croisement. Les propriétés de convergence des algorithmes génétiques sont donc fortement dépendantes de cet opérateur.

Cet opérateur de mutation est utilisé avec une probabilité (P_m) nommée probabilité de mutation. Dans les algorithmes génétiques à codage binaire. Cette probabilité s'effectue sur les gènes en échangeant sa valeur de 0 à 1 et non sur le chromosome tout entier.

Tableau 5 : Opérateur de mutation d'un bit

Parent	0	1	1	0	1	0	0	Enfant	0	1	0	0	1	0	1
--------	---	---	---	---	---	---	---	--------	---	---	---	---	---	---	---

Mais avec un algorithme génétique codé autrement, on applique cette probabilité par rapport à l'individu tout entier et non sur les gènes. Si β , généré aléatoirement, appartient à $[0, P_m]$, nous appliquons l'opérateur de mutation sur cet individu.

1.5.6 Méthode d'insertion

Après l'étape de mutation, on utilise une méthode d'insertion pour générer une nouvelle population. Lors de la construction de cette population, on se trouve devant un vrai problème : faut-il garder les enfants ou les parents ou bien un certain pourcentage des deux en respectant que la taille de la population (N) reste constante ?

Il s'agit de concevoir une stratégie d'évolution de la population. Nous distinguons dans la littérature deux stratégies :

- La première stratégie, notée (N, N_f), consiste à choisir les N individus à partir de N_f enfants déjà créés par les opérateurs de croisement et mutation. Dans cette stratégie,

⁴ Cerf, R. "Une théorie asymptotique des algorithmes génétiques". Thèses de doctorat, Université MontpellierII.1994.France.

on suppose que $N_f \geq N$. Quand $N_f = N$, nous parlerons de la méthode générationnelle qui remplace les parents par les enfants.

- La seconde, notée $(N + N_f)$, consiste à choisir les N individus à partir des N parents de la population précédente et de N_f nouveaux enfants. Un cas particulier de cette stratégie, appelé méthode d'état d'équilibre, a pour principe de sauvegarder une grande partie de la population dans la génération suivante. Coûts seront sélectionnés afin de créer des chromosomes fils qui remplaceront les plus mauvais parents. Le reste de la population survie et sera copié dans la nouvelle génération.

L'élitisme est une stratégie complémentaire de la première stratégie. Il consiste à copier quelques meilleurs chromosomes dans la nouvelle population. Il accroît l'efficacité de l'algorithme génétique basé sur la méthode d'insertion générationnelle. L'objectif est d'éviter que les meilleurs chromosomes soient perdus après les opérations de croisement et de mutation. Cette méthode améliore considérablement les algorithmes génétiques, car elle permet de conserver à une itération k , le meilleur individu trouvé dans toutes les populations générées antérieurement.

1.5.7 Test d'arrêt :

Le test d'arrêt joue un rôle primordial dans le jugement de la qualité des individus son but est de nous assurer l'optimalité, de la solution finale obtenue par l'algorithme génétique.

Les citées d'arrêts sont de deux natures :

1. Arrêt après un nombre fixé a priori de générations. C'est la solution retenue lors qu'une durée maximale de temps de calcul est imposée.
2. Arrêt lorsque la population cesse d'évoluer ou n'évolue plus suffisamment. Nous sommes alors en présence d'une population homogène dont on peut penser qu'elle se situe à la proximité de l'optimum. Ce test d'arrêt reste le plus objectif et le plus utilisé.

Il est à notre qu'aucune certitude concernant la bonne convergence de l'algorithme n'est assurée. Comme dans toute procédure d'optimisation l'arrêt est arbitraire, et la solution « en temps fini » ne constitue qu'une approximation de l'optimum.

1.6 Conclusion :

Un algorithme génétique vous donne une grande liberté dans le paramétrage et dans l'implémentation des différents traitements. Libre à vous ensuite de modifier tel ou tel paramètre si les solutions obtenues ne vous conviennent pas.

Les algorithmes génétiques ont l'énorme avantage de pouvoir être appliqués dans un grand nombre de domaines de recherche de solution, lorsqu'il n'est pas nécessaire d'avoir la solution optimale, qui prendrait par exemple trop de temps et de ressources pour être calculée (ou tout simplement si personne n'est capable de la trouver de manière théorique).

Nous allons passer au chapitre suivant qui contient les problèmes réels par domaine : Domaine de transport aérien et les problèmes de tournées de véhicule.

Chapitre 02

Problèmes réels par domaine

2.1 Introduction

Les algorithmes génétiques (AGs) sont basés sur les phénomènes de reproduction de la nature qui mettent en œuvre : le croisement, la mutation et la sélection. Ils permettent, par un codage approprié, de trouver en un temps limité, de bonnes solutions aux problèmes d'optimisation complexes. Dans ce chapitre on va traiter les Problèmes réels par domaine (Domaine de transport aérien et Les problèmes de tournées de véhicules)

2.2 Domaine de transport aérien : Gestion du roulage sur les grands aéroports

Le développement récent de nombreux hubs commerciaux est à l'origine d'une nouvelle forme de congestion sur les grandes plates-formes aéroportuaires : la plupart des mouvements ont tendance à être programmés aux mêmes heures « stratégiques ».

Les nombreux retards occasionnés sont la source d'une incertitude croissante sur les heures de décollage et d'atterrissage. Les retards peuvent atteindre plusieurs dizaines de minutes aux heures de pointe, ce qui est extrêmement pénalisant pour l'ensemble des acteurs du transport aérien. Cette application présente une modélisation du trafic au roulage sur un aéroport. Elle est intégrée dans le simulateur décrit dans la première partie.

Différentes méthodes d'optimisation sont comparées par corrélation entre le délai qu'elles génèrent et le nombre de mouvements impliqués. Elles utilisent des algorithmes génétiques et des algorithmes de parcours de graphe pour trouver le meilleur chemin et/ou les meilleurs points d'attente pour chaque avion, en respectant les normes de séparation au sol, pistes comprises.

2.2.1 Modélisation

Le problème est de trouver un ensemble optimal de trajectoires admissibles pour le trafic au roulage.

Une trajectoire est définie par une heure de départ (ou d'arrivée), un chemin et des points d'attente sur ce chemin.

Le caractère optimal de l'ensemble des trajectoires peut avoir différentes définitions,

et sera considéré globalement comme le minimum d'une fonction de cout précisée ci-dessous.

Les trajectoires sont admissibles si d'une part les chemins empruntés sont conformes avec les contraintes opérationnelles de l'aéroport, et si d'autre part les avions sont séparés (comme détaillé dans les parties suivantes).

2.2.2 AG : algorithmes génétiques

Une stratégie de résolution est développée et utilise des algorithmes génétiques classiques^{5 6}

Cette stratégie recherche un chemin et éventuellement une attente par mouvement.

2.2.2.1 Codage des données

La trajectoire d'un avion est décrite par 3 paramètres : le numéro n du chemin suivi, la position p d'attente, et l'heure t de fin d'attente (si p est atteinte après l'heure t , l'avion n'attend pas). Les éléments de la population (ou chromosomes) sont donc constitués de $3N$ variables pour un problème à N avions.

2.2.2.2 Fonction d'adaptation

Les trajectoires décrites par un chromosome peuvent ne pas être admissibles. La fonction d'adaptation (ou fitness) F à maximiser, comprise entre 0 et 1, doit assurer qu'un chromosome représentant une solution admissible est toujours mieux évalué qu'un chromosome décrivant une situation avec conflits.

Pour cela, la fitness d'une solution admissible sera toujours supérieure à 1 et la Fitness des éléments contenant des conflits inférieure à 2.

Si n_c est le nombre de conflits et f_c la fonction de coût décrite au 2.2, la fitness sera

⁵ D. E. Goldberg. "Algorithmes génétiques, exploration, optimisation et apprentissage automatique". Addison Wesley, 1995. Paris France.

⁶ Z Michalewicz. "Genetic algorithms + Data Structures = Evolution Programs". Springer-verlag, 1992 Mishawaka, IN, U.S.A.

donnée par :

$$\begin{aligned} \text{si } n_c > 0, \quad F &= \frac{1}{1+n_c} \\ \text{si } n_c = 0, \quad F &= \frac{1}{2} + \frac{1}{1+f_c} \end{aligned}$$

2.2.2.3 Croisement et Mutation

Le caractère partiellement séparable du problème de résolution de conflits est à nouveau utilisé.

Un fitness local F_i est calculée pour chaque avion i , en fonction du nombre de conflits n_{c_i} impliquant cet avion et de sa participation f_{c_i} à la fonction de coût (cf. 2.2) :

$$\text{si } n_{c_i} > 0, F_i = K n_{c_i} \quad \text{sinon, } F_i = f_{c_i}$$

(K étant une constante telle que $K \gg f_c$)

2.2.2.4 Sharing

Le problème est combinatoire et présente de nombreux optima locaux. Le sharing décrit dans Yin et Germy⁷ permet d'empêcher la population de s'homogénéiser trop rapidement sur les optima locaux.

Il nécessite l'introduction d'une distance entre deux éléments A et B de la population, cette distance étant utilisée pour dissocier différents groupes (clusters) dans la population. La distance définie est la suivante :

$$D(A, B) = \frac{\sum_{i=1}^N |l_{A_i} l_{B_i}|}{N}$$

l_{A_i} (resp l_{B_i}) étant la longueur du chemin de l'avion i dans le chromosome A (resp B).

⁷ Xiaodong Yin and Noel Germy. "A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimi" Springer-Verlag, 1993. U.S.A.

2.2.2.5 Critère d'arrêt

Le critère d'arrêt de l'algorithme génétique est défini par un nombre maximal absolu de 50 générations, et un nombre maximal de 20 générations sans conflit (i.e. à meilleur élément décrivant une solution admissible).

2.2.2.6 Résolutions par Clusters

Pour diminuer autant que possible la complexité du problème, une fermeture transitive est appliquée aux paires d'avions en conflit dans la situation initiale (sans attente ni déviation). Les ensembles (ou clusters) d'avions ainsi formés sont résolus séparément.

A chaque fois que la résolution séparée de deux clusters provoque des conflits entre leurs avions, les deux clusters sont réunis et une nouvelle optimisation est effectuée.

2.3 Les problèmes de tournées de véhicules

Le problème de tournées de véhicules (VRP) est un problème majeur de l'optimisation combinatoire. Il a fait l'objet de plusieurs études motivées par la difficulté de sa résolution et par ses nombreuses applications pratiques en logistique. Avec l'essor des méta-heuristiques telles que la recherche taboue et les algorithmes génétiques, plusieurs instances de problèmes ont pu être résolues de manière efficace, voire optimale. Cependant, les méthodes de résolution utilisées sont souvent entièrement automatiques et ne font intervenir l'humain que dans la phase de modélisation du problème. Elles ne permettent pas ainsi d'exploiter toute la richesse de la connaissance et de l'expertise que des opérateurs humains peuvent avoir. De plus, leur fonctionnement en « boîte noire » ne répond pas au besoin de ces opérateurs de comprendre le cheminement du processus d'optimisation.

2.3.1 L'algorithme génétique interactif pour le VRP

Un algorithme génétique interactif pour la résolution de problèmes de tournées de véhicules. Dans le processus d'optimisation, les tâches sont clairement réparties entre l'humain et la machine : le système affiche graphiquement les solutions de la population courante l'humain les inspecte visuellement et exprime des préférences de type « j'aime » ou « je n'aime pas » en spécifiant les parties qu'il juge intéressantes ou non intéressantes. Avant de générer une nouvelle population, ces annotations humaines (tags) sur les phénotypes sont

traduites sur les génotypes pour pouvoir être prises en compte par l’algorithme génétique.

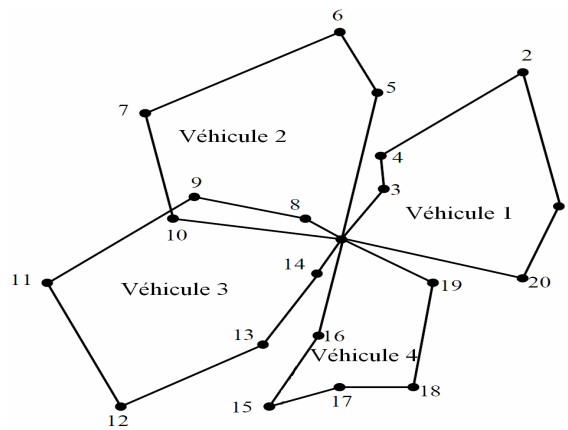


Figure2 : Solution S d’un VRP

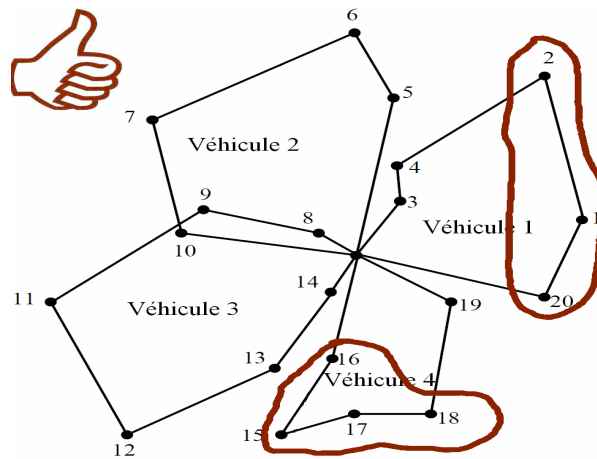


Figure3 : Solution S annotée

La figure 2 montre un exemple de VRP avec $n = 20$ clients résolu avec $m = 4$ véhicules, et la figure 3 montre une annotation de cette solution. En utilisant un codage des solutions sous forme de permutations de clients sans séparateurs de tournées⁸, les deux solutions précédentes sont respectivement représentées par les deux chromosomes suivants :

20	1	2	4	3	5	6	7	10	8	9	11	12	13	14	16	15	17	18	19
20	1	2	4	3	5	6	7	10	8	9	11	12	13	14	16	15	17	18	19

⁸ C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research, 31(12) :1985–2002, 2004.

Les cellules en gris dans le deuxième chromosome correspondent aux deux parties de la solution jugée intéressantes par l'humain. Un tel tag a une incidence sur les trois étapes clés de l'algorithme, à savoir, la sélection, les croisements et les mutations.

En effet, cette solution marquée intéressante se voit attribuer une probabilité de sélection plus grande. Si elle est sélectionnée comme parent dans un croisement, les deux indices de croisement (opérateur Order Crossover (OX)) sont tirés aléatoirement en dehors des intervalles grisés pour sauvegarder les séquences de clients jugées intéressantes par l'humain. Il en est de même pour l'opérateur de mutation qui n'est pas autorisé à modifier les parties taguées.

Avec un système d'affichage graphique et un ensemble d'interactions simples et intuitives, un humain peut ainsi exprimer des préférences sur les solutions générées par un algorithme génétique et diriger la convergence de celui-ci tout en la comprenant.

2.4 Conclusion

Dans ce chapitre nous avons abordé le problème de transport aérien et le problème de tournées de véhicules. La plupart des mouvements sont programmés aux mêmes heures, ce qui cause des retards. Nous avons aussi réalisé une analyse du domaine de travail pour le problème de tournées des véhicules. Nous présenterons dans le chapitre suivant le problème de voyageur de commerce.

Chapitre 03

Problème du voyageur de commerce

3.1 Introduction

Dans ce chapitre nous allons exposer et définir le problème de voyageur de commerce, puis nous présenterons ses intérêts et les domaines d'application qui peuvent être appliqués à ce problème et nous finirons par un bref historique sur le problème de voyageur de commerce.

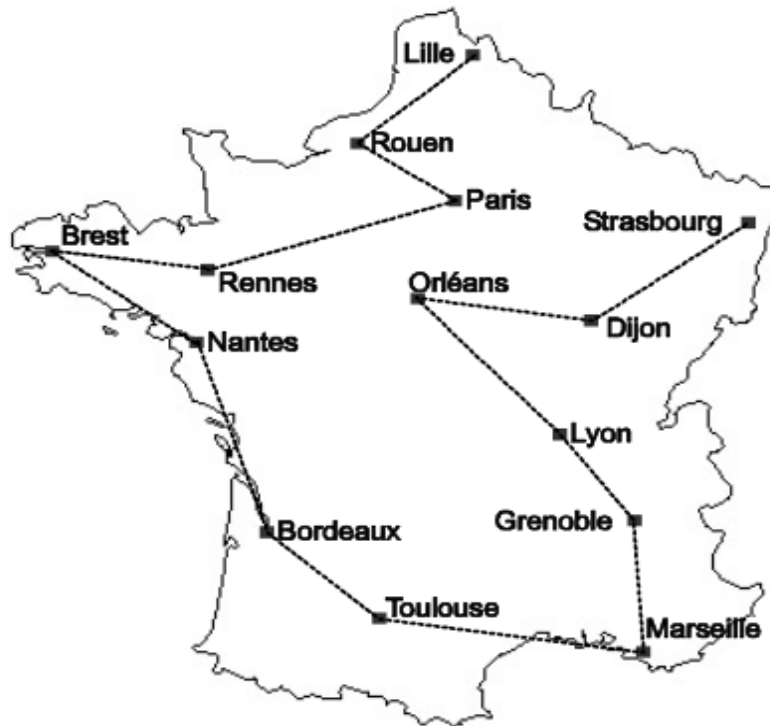


Figure4 : voyageur de commerce (la carte du France)

Le problème du voyageur de commerce étant combinatoire (le nombre de solutions est de l'ordre $n!$, n nombre de villes), le but du projet est d'appliquer des algorithmes génétiques pour trouver le meilleur circuit de coût minimal. La mise-en-œuvre de cet algorithme consiste particulièrement à intégrer des nouveaux opérateurs de sélection et de croisement génétique.

3.2 Théoriquement

Soit un graphe $G = (S, A)$, le problème est de trouver un parcours qui passe par tous les sommets une et une seule fois et qui soit de poids minimal. En général, on travaille sur un graphe complet.

3.2.1 Pratiquement

Un voyageur de commerce doit visiter N villes données en passant par chaque ville exactement une fois. Il commence par une ville quelconque et termine en retournant à la ville de départ. Les distances entre les villes sont connues. Quel chemin faut-il choisir afin de minimiser la distance parcourue? La notion de distance peut-être remplacée par d'autres notions comme le temps qu'il met ou l'argent qu'il dépense : dans tous les cas, on parle de coût.

3.2.2 Complexité du Problème

La complexité du problème croît comme la factorielle du nombre de villes. Si on considère des permutations simples, il existe 3 628 000 permutations de 10 villes. Pour 100 villes, on passe à 10 puissances 158. La complexité du problème croît de manière plus que polynomiale (problème dit NP).

Tableau 6 : Tableau de Complexité du problème

N	Nombre de possibilités	Temps de ca
5	12	Microseconde
10	181440	Deuxième seconde
15	43 milliards	Dizaine d'Heures
20	$60 \cdot 10^{15}$	Milliers d'Années
25	$310 \cdot 10^{21}$	Milliards d'Années

3.2.3 Intérêt

Le PVC fournit un exemple d'étude d'un problème NP-complets dont les méthodes de résolution peuvent s'appliquer à d'autres problèmes de mathématiques discrète. Néanmoins, il a aussi des applications directes, notamment dans les transports et la logistique. Par exemple, trouver le chemin le plus court pour les bus de ramassage scolaire ou, dans l'industrie, pour trouver la plus courte distance que devra parcourir le bras mécanique d'une machine pour percer les trous d'un circuit imprimé (les trous représentent les villes).

3.3 Application de l'AG au PVC

3.3.1 L'Objectif

L'objectif est d'intégrer l'algorithme génétique avec les opérateurs de sélection et de croisement pour résoudre le problème du voyageur de commerce⁹ ¹⁰.

3.3.2 Les Statistiques sur les Paramètres de L'AG

Comme il y a plusieurs types d'opérateurs, et des paramètres, nous préférons bien savoir quelle est les différences de l'un et l'autre d'opérateur, et des choix de paramètre pour un résultat obtenu. Afin de voir la performance et l'efficacité de chaque algorithme, on fait une statistique avec la distance finale et le temps d'exécution sur le même problème.

Tableau 7 : Tableau de Ststatistique sur les Paramètres de l'AG

Nombre de ville	Croisement	Probabilité de croisement	Sélection	Probabilité de mutation	Distance Moyenne	Ecart Type
20	CPA	100%	Roulette	0.1	121.4	2253
20	CPA	60%	Roulette	0.1	99.5	2136
20	CPA	100%	Roulette	0.02	151.0	2248
20	CPA	100%	Roulette	0.2	102.2	2244
20	1X	100%	Roulette	0.1	101.9	2159
20	0X	100%	Roulette	0.1	36.7	2068
20	CPA	100%	Rang	0.1	133.0	2289
20	CPA	100%	Tournoi	0.1	193.2	2322

⁹ https://www.gerad.ca/Sebastien.Le.Digabel/MTH6311/8_applications_2.pdf

¹⁰ Thèse de Doctorat de l'Université du Havre .Thèse Jean Philippe Vacher.pdf .2004. Valenciennes France.

3.4 Conclusion

Le problème du voyageur de commerce est toujours d'actualité dans la recherche en informatique, étant donné le nombre important de problèmes réels auxquels il correspond. Les problèmes dérivés et les extensions sont très nombreux. Par exemple, des fenêtres de temps peuvent y être ajoutées. Ce concept consiste à imposer des contraintes de temps pour la traversée de chaque sommet. Autre exemple, il peut y avoir plusieurs voyageurs de commerce partant d'un même sommet, ou de sommets différents. Il suffit alors de considérer que les voyageurs de commerce sont des véhicules pour arriver à des problèmes de tournées de véhicules : étant donnée une flotte de véhicules, le problème consiste à déterminer les trajets de chacun pour livrer à moindre coût des clients en marchandise (chaque client est représenté par un sommet dans le graphe). Le nombre de véhicules peut être fixe ou non, les capacités des véhicules peuvent être les mêmes ou non, des fenêtres de temps peuvent être définies etc. Pour chacune de ces variantes, de nouvelles méthodes peuvent être explorées.

Dans ce qui suit nous allons présenter les différentes étapes de conception et d'implémentation de notre système.

Chapitre 04

Conception et Implémentation

4.1 Introduction

Dans ce chapitre nous abordons les aspects de conception de notre système, en montrant les différents diagrammes de conception, qui décrivent son architecture et son fonctionnement. Notre système concerne l'application des algorithmes génétiques. Pour la résolution du problème du voyageur du commerce.

4.2 Conception

Nous avons modélisé les besoins de système des algorithmes génétiques, en se basant sur le langage de modélisation UML «Unified Modeling Language». Nous avons choisi pour cette modélisation trois diagrammes principaux, le diagramme de cas d'utilisation et le diagramme d'activité, et le diagramme classe qui représentent les besoins fonctionnels ainsi que l'architecture du système.

Nous avons utilisé comme outil de modélisation UML, l'outil StarUML qui est un logiciel de modélisation, qui aide à tracer les différents diagrammes UML. Cet outil donne la possibilité d'exporter les diagrammes réalisés en images de format «png» ou «jpg» afin de les insérer au sein du document. La Figure montre l'interface principale de l'outil de modélisation StarUML version 3.1.0.

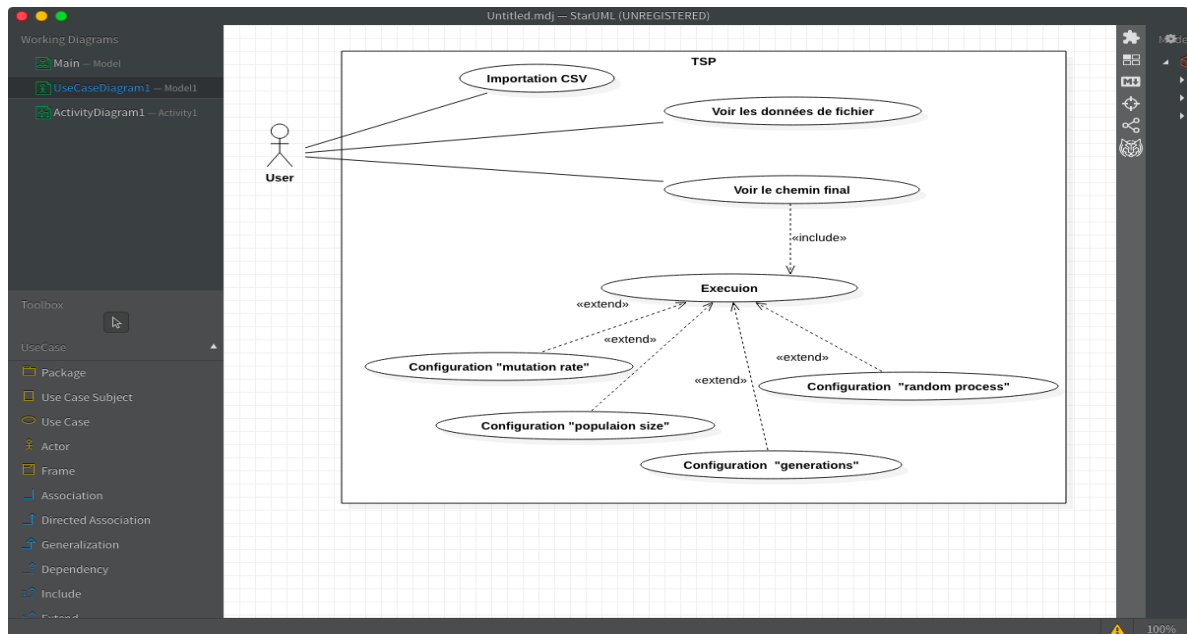


Figure 5 : Interface de StarUML version 3.1.0

4.2.1 Diagramme de cas d'utilisation

Nous avons commencé la modélisation par le diagramme de cas d'utilisation, qui permet de recueillir, d'analyser et d'organiser les besoins, et de recenser les grandes fonctionnalités du système.

La figure 6 représente le diagramme de cas d'utilisation générale qui fait preuve d'un acteur appelé utilisateur qui admet les fonctionnalités suivantes :

- Importation CSV ;
- Voir les données de fichier ;
- Voir le chemin final ;

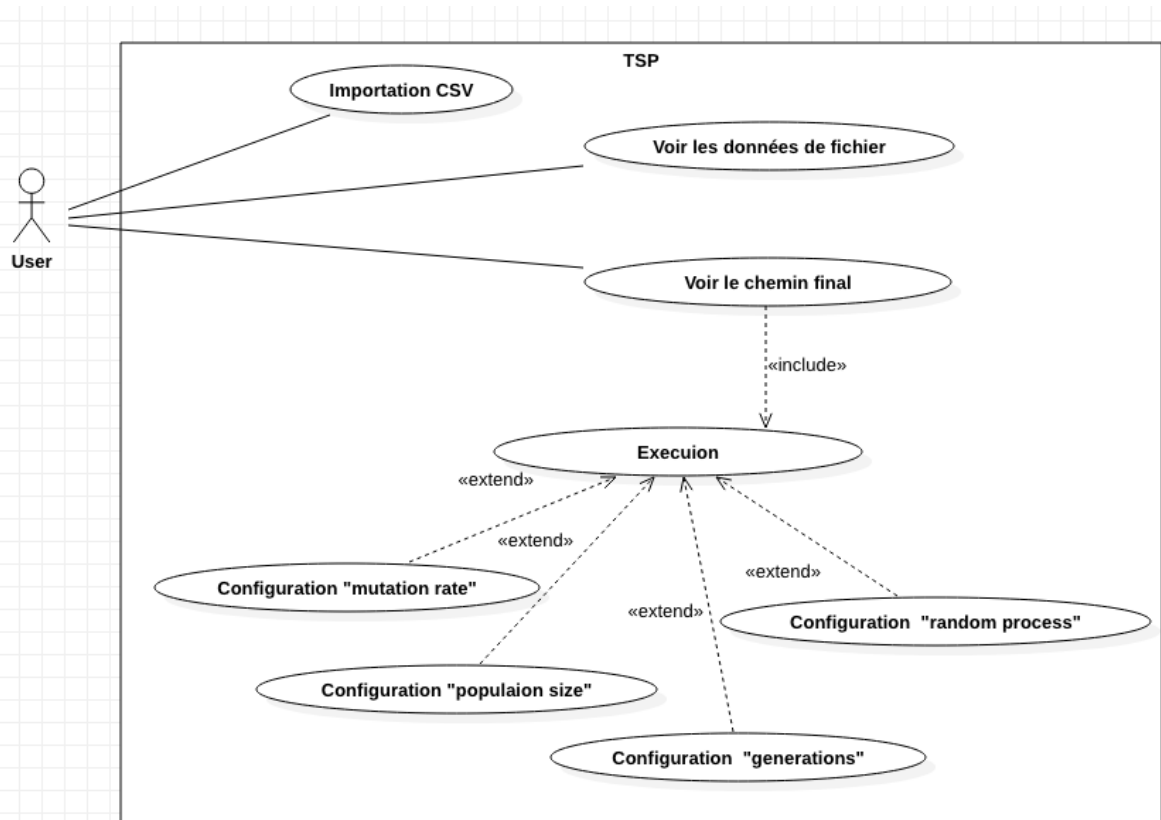


Figure 6 : Diagramme de cas d'utilisation

4.2.2 Diagramme d'activité

Le diagramme d'activité permet de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôles et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

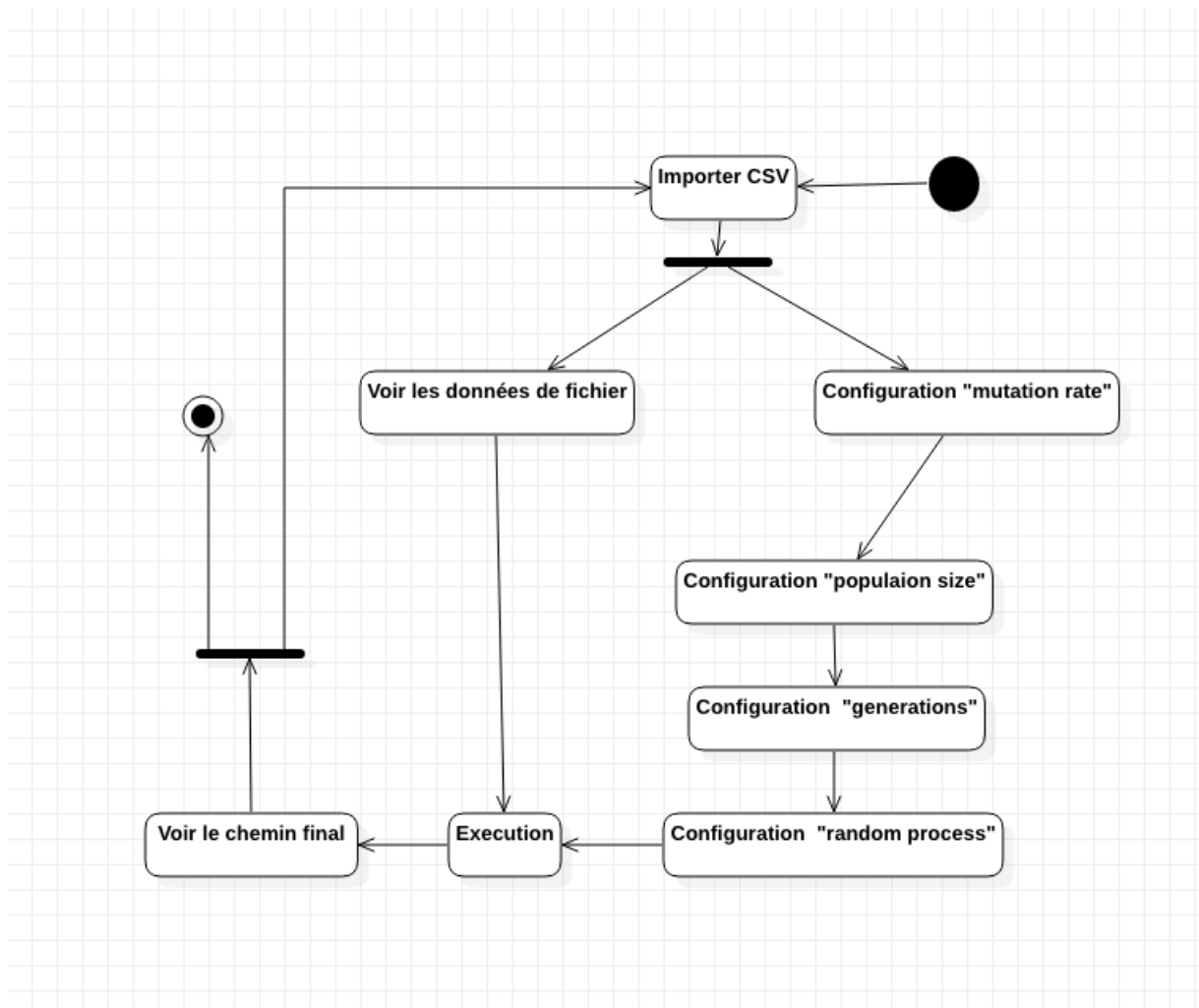


Figure7 : Diagramme d'activité

4.2.3 Diagramme de classe

Le diagramme de classe est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Le diagramme de classe du système de notre projet est représenté comme suit :

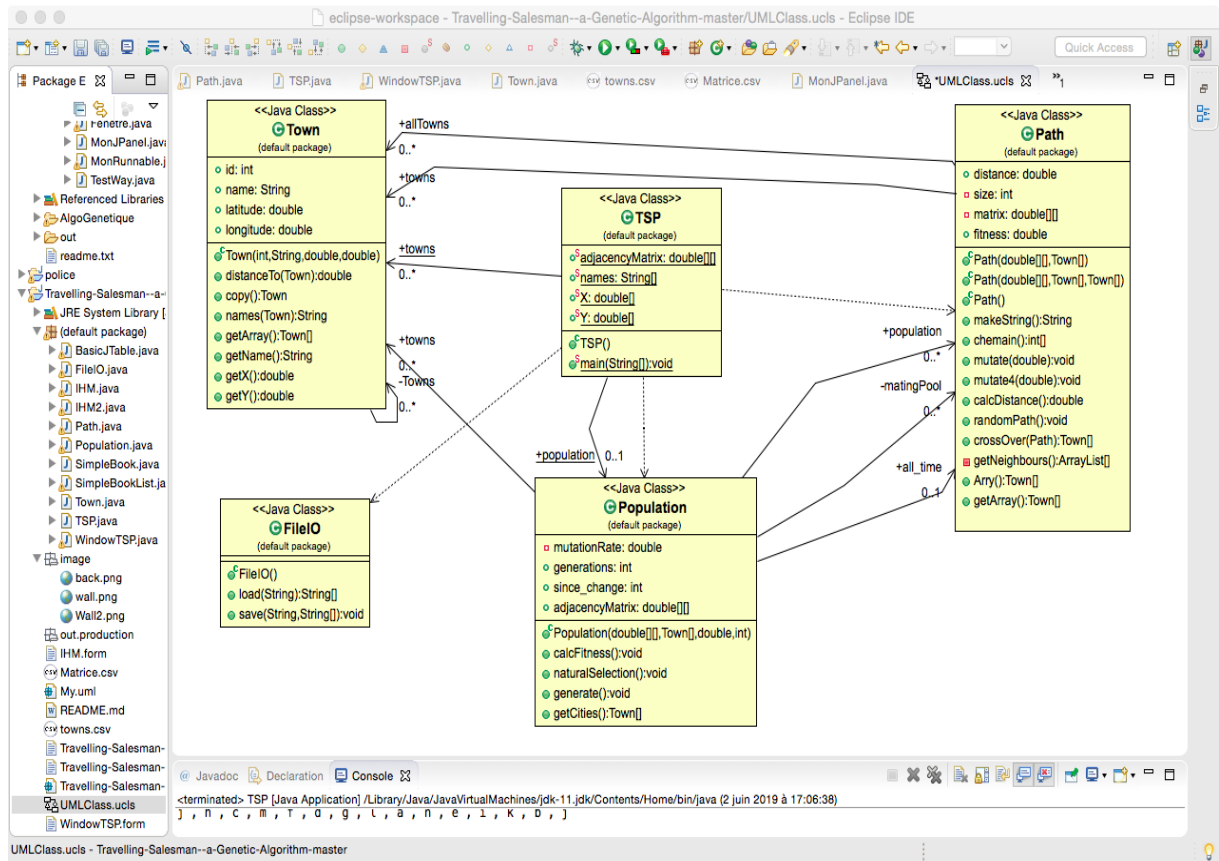


Figure 8 : Diagramme de classe

4.3 Implémentation

4.3.1 Les coordonnées géographiques

Les coordonnées géographiques sont formées à partir des deux composantes que sont la latitude et la longitude.

La position nord-sud d'un point est donnée par sa latitude, quand sa longitude nous renseigne sur sa position est-ouest.

4.3.1.1 Définition de la latitude

La latitude d'un point est la mesure de l'angle formé par le plan de l'équateur avec la droite reliant le centre de la Terre au point en question.

Par construction, elle est donc comprise entre -90° et 90° , les valeurs négatives étant pour l'hémisphère sud, et elle vaut 0° à l'équateur.

4.3.1.2 Définition de la longitude

Le principe est le même pour la longitude, à la différence près qu'il n'existe pas de référentiel naturel comme l'équateur. La longitude de référence a donc été fixée arbitrairement au méridien de Greenwich (il passe par l'Observatoire royal de Greenwich, à Greenwich dans la banlieue de Londres), et la longitude d'un point est la mesure de l'angle formé par le demi-plan formé par l'axe de la Terre et passant par le méridien de Greenwich et le demi-plan formé par l'axe terrestre et passant par le point en question.

4.3.1.3 Les différents référentiels de coordonnées Géographiques :

Les définitions précédentes prennent en compte plusieurs paramètres qui doivent être fixés ou identifiés pour servir de référence:

- le plan de l'équateur et la modélisation de la forme de la Terre retenue,
- un ensemble de points de référence,
- la position du centre de la Terre,
- l'axe de la Terre,
- le méridien de référence.

Ces cinq critères sont à la base des différents systèmes géodésiques utilisés au cours de l'histoire.

A l'heure actuelle, le plus couramment utilisée est le système WGS 84 (utilisé notamment pour les coordonnées GPS).

4.3.2 Environnement matériel

Pour le développement, un ordinateur portable MacBook avec les caractéristiques suivantes:

1. Processeur : Intel Core 2 Duo
2. RAM : 4 GO

4.3.3 Environnement logiciel

1. Système d'exploitation : Mac OS X Yosemite 64-bits.
2. Outil de modélisation: StarUML version 3.1.0
3. Outil de développement: Eclipse IDE, NetBeans IDE, IntelliJ IDEA

4.3.4 Langage et outils de développement

Nous avons utilisé un langage et plusieurs outils pour le développement de notre système. Nous avons utilisé le langage et les outils comme suit:

- Nous avons utilisé l'outil Eclipse IDE et NetBeans IDE et IntelliJ IDEA pour réaliser les interfaces graphiques de notre système.
- Nous avons utilisé le langage de programmation Java¹¹,

4.4 Les interfaces graphiques principales

Dans cette partie nous allons présenter les différentes parties de notre système, en montrant les interfaces graphiques principales qui représentent quelques fonctionnalités du système.

¹¹ <https://www.java.com>

- La Figure 9 montre l'interface graphique principale pour importer un fichier .csv :

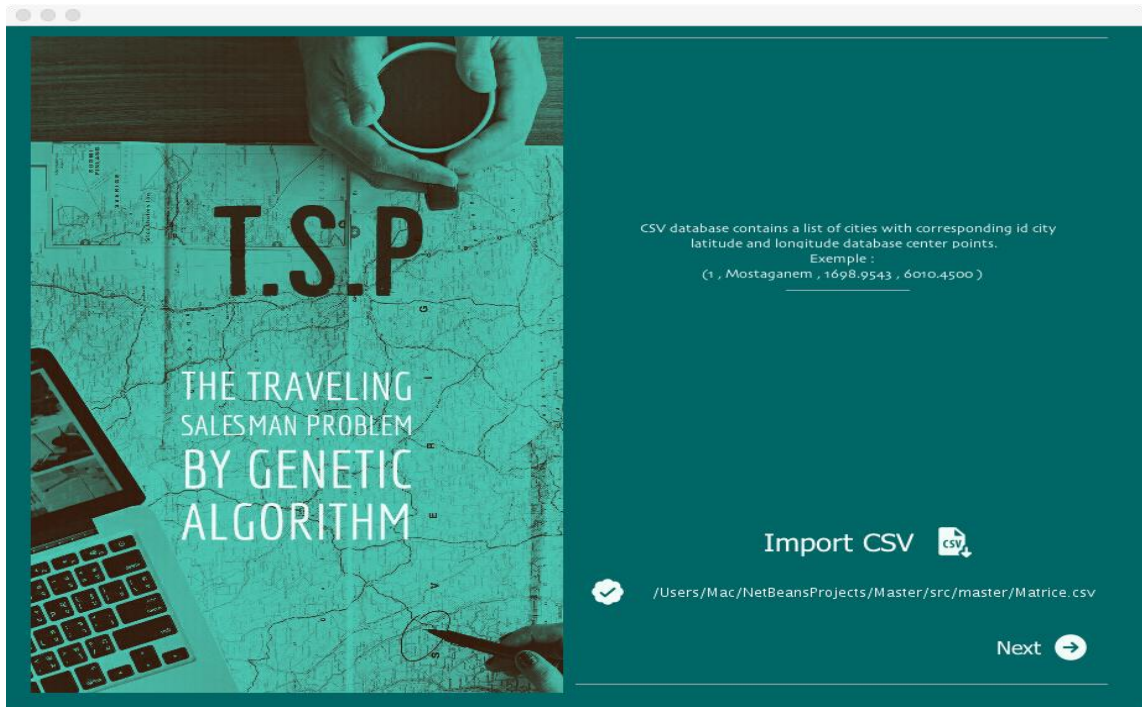


Figure 9 : l'interface graphique principale pour importer un fichier .csv

- La Figure 10 montre la deuxième interface graphique qui contient la présentation du fichier csv dans sa forme originale et s'affiche sous forme de courbe.

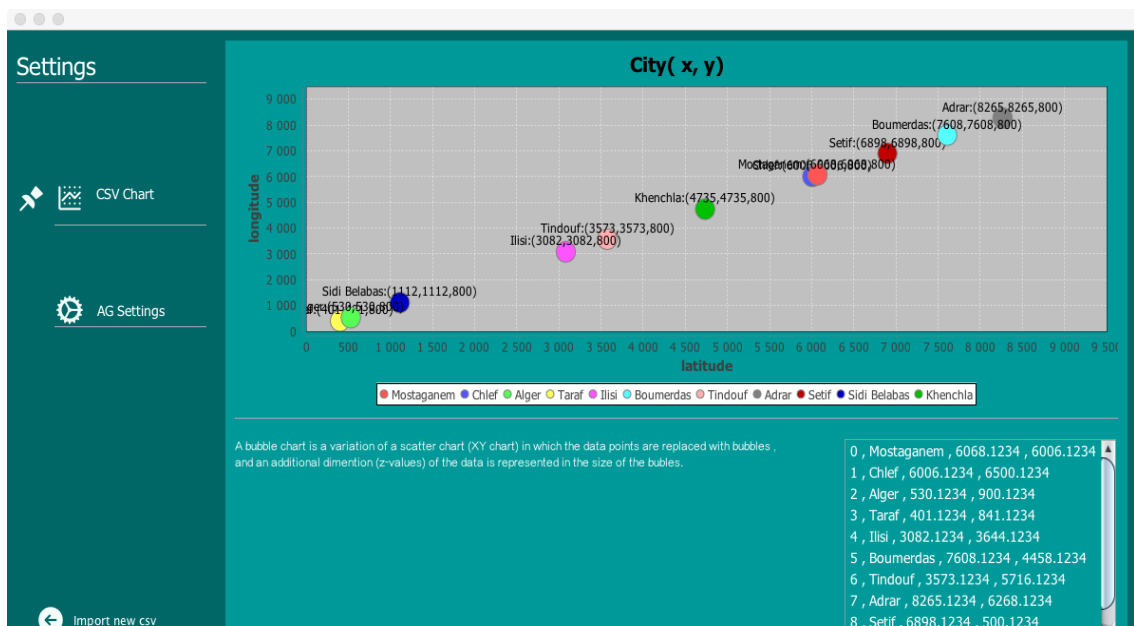


Figure 10: l'interface graphique de fichier csv dans sa forme originale

- La Figure 11 montre la troisième interface graphique de paramètres des algorithmes génétiques qui sont saisis par défaut: Mutation rate, Population size, Génération, Random process

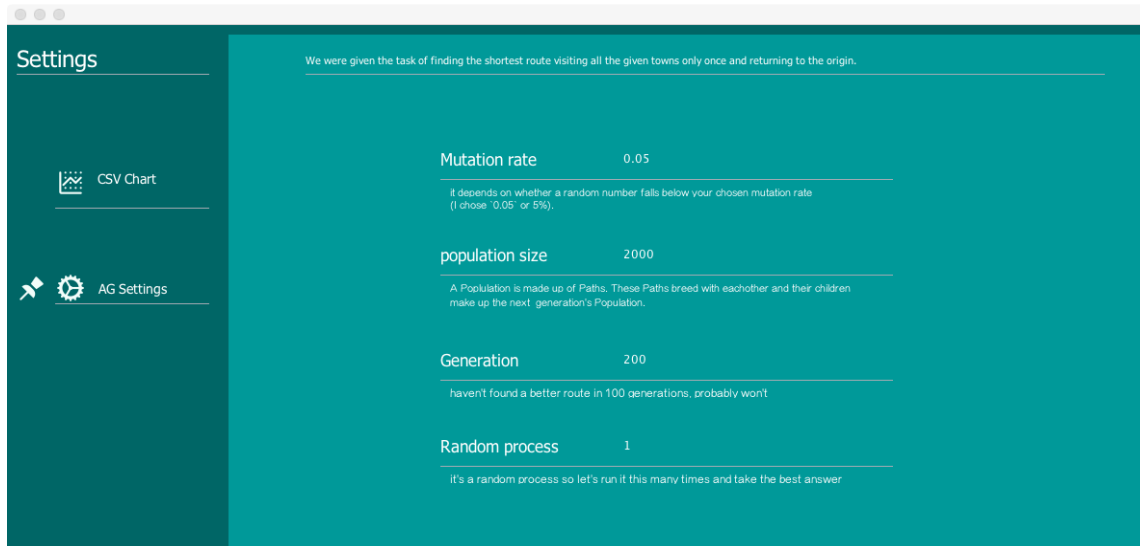


Figure 11: interface graphique de paramètres des algorithmes génétiques

- La Figure 12 montre la quatrième interface graphique de l'exécution finale : distance initiale, distance finale (meilleure distance) et le temps de calcul.

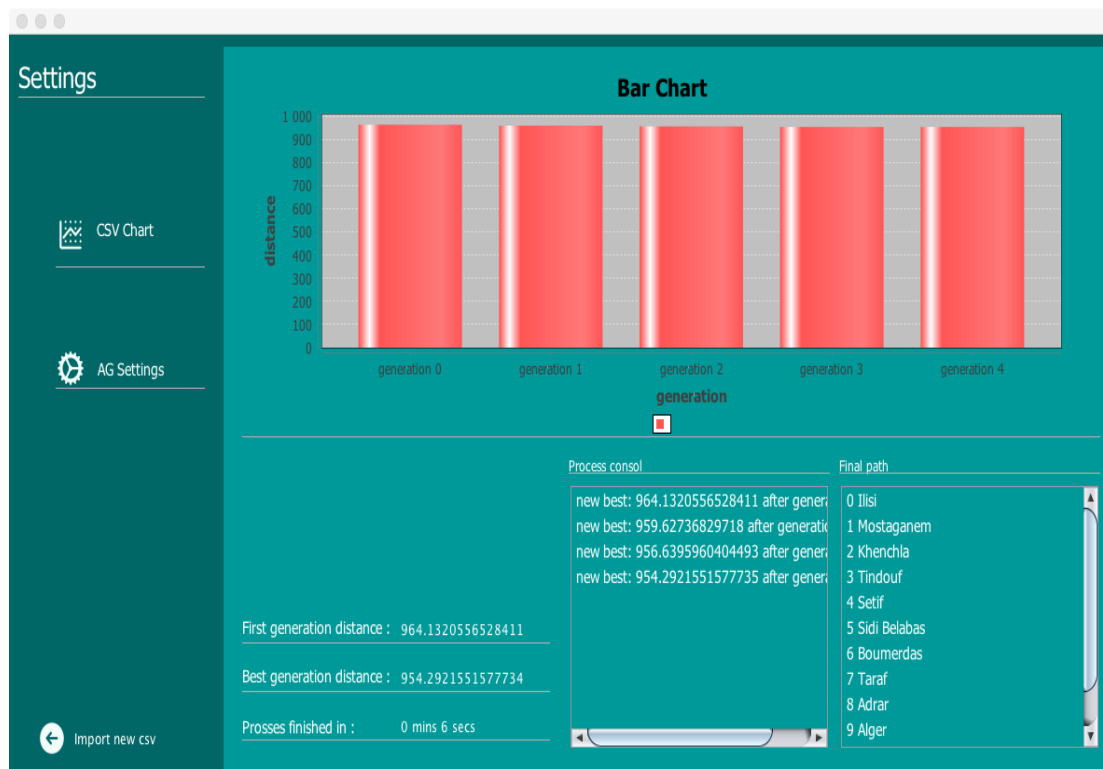


Figure 12: interface graphique de chemin final

4.5 Conclusion

A travers ce chapitre nous avons pu présenter les différentes étapes de conception de notre système, en montrant les différents diagrammes UML utilisés. Nous avons présenté l'environnement de travail, et détaillé ensuite le langage les différents et outils utilisés pour le développement de notre système. Nous avons terminé par une présentation des interfaces graphiques principales de fonctionnement du système.

Conclusion générale

Conclusion générale

A l'issue de ce travail, nous prenons conscience du fabuleux potentiel des algorithmes génétiques. Leurs applications sont multiples : optimisation de fonctions numériques difficiles, traitement d'image, optimisation d'itinéraires et d'emplois du temps, design industriel, etc. Beaucoup d'aspects nous ont séduits dans ces algorithmes.

Ce travail nous a permis de connaître la mise en œuvre de l'algorithme génétique qui est largement utilisé dans la recherche scientifique.

Dans notre travail nous avons appliqué les algorithmes génétiques au problème du voyageur de commerce, c'est-à-dire trouver le meilleur circuit au coût minimal. Ceci nous a permis de nous rendre compte que le codage des données pour modéliser un problème est complexe. D'autre part, nous nous sommes aussi aperçus des difficultés pour choisir pertinemment de bons paramètres pour les divers opérateurs (mutation, croisement, sélection, remplacement). Des choix par rapport aux opérateurs eux-mêmes sont aussi à gérer, sachant que certains sont plus appropriés au problème et qu'ils permettent d'optimiser.

Bibliographie

- [1] Holland, J. H. "Adaptation in Natural and Artificial Systems". Ann Arbor : University of Michigan Press. 1975. Libraries Australia.183 pages.
- [2] Goldberg, D. "Genetic Algorithm In Search, Optimization And Machine Learning". Addison-Wesley Longman . 1989.Boston, USA.313 pages.
- [3] Lerman, I. &Ngouenet, F. "Algorithmes génétiques séquentiels et parallèles pour une représentation affine des proximités".Rapport de Recherche de l'INRIA.1995. Rennes France. Projet REPCO 2570.
- [4] Cerf, R. "Une théorie asymptotique des algorithmes génétiques". Thèses de doctorat, Université MontpellierII.1994.France.
- [5] D. E. Goldberg. "Algorithmes génétiques, exploration, optimisation et apprentissage automatique". Addison Wesley, 1995. Paris France.
- [6] Z Michalewicz. "Genetic algorithms + Data Structures = Evolution Programs". Springer-verlag, 1992 Mishawaka, IN, U.S.A.
- [7] Xiaodong Yin and Noel Germay. "A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimi" Springer-Verlag, 1993. U.S.A.
- [8] C. Prins. "A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research, 31(12) :1985–2002, 2004.
- [10] Thèse de Doctorat de l'Université du Havre .Thèse Jean Philippe Vacher.pdf .2004. Valenciennes France.367 pages.

Webographie

[9] Résolution de problème Voyageur de commerce :

https://www.gerad.ca/Sebastien.Le.Digabel/MTH6311/8_applications_2.pdf

[11] <https://www.java.com>