



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : IA /M2/2019

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Intelligence artificielle**

Un algorithme évolutionnaire inspiré de l'informatique quantique pour les problèmes discrets

Par :

GOUCEM MAROUA

Soutenu le 14 septembre 2020, devant le jury composé de :

BERGHIDA Meryem

M.C.B

Président

Rapporteur

Examineur

Dédicace

Je dédie ce mémoire :

A la mémoire de mon père

Qui m'a toujours été l'épaule solide, mon exemple éternel, ma force et la personne la plus digne de mon amour et de mon estime et de mon respect.

J'espère que, du monde qui est sien maintenant, il apprécie cet humble geste comme preuve de reconnaissance de la part d'une fille qui a toujours prié pour lui.

A ma chère mère

Tu m'as donné la vie, la tendresse, le courage pour réussir toutes les années de mes études.

Tu as fait plus qu'une mère puisse faire pour que ses enfants suivent le bon chemin dans leur vie et leurs études. Tout ce que je peux t'offrir ne pourra exprimer l'amour et la reconnaissance que je te porte pour te remercier pour tes sacrifices. Puisse Dieu te donner santé, bonheur et longue vie.

A mes chères sœurs

Fatima, ma grande sœur, la tendresse, au cœur grand, **Kenza**, la prune de mes yeux, **Faiza**, l'aimable, qui a toujours été de mon côté, **Manar**, ma bougie, que j'aime profondément.

Tous les mots ne sauraient exprimer la gratitude, l'attachement, l'amour et l'affection que je porte pour vous. Je vous souhaite une vie pleine de bonheur et de succès et que Dieu, vous protège et vous garde.

A mes chers neveux

Louay, **Racim**, mes sources de bonheur. Vous avez apporté beaucoup de bonheur à notre famille. Puisse Dieu vous garder et éclairer votre route. Je vous aime.

A ma meilleure amie

Oumaima Anfel, symbole de fidélité. Vous êtes pour moi ma sœur sur qui je peux compter. Je suis fière de l'amitié qui nous unit et des souvenirs de tous les moments que nous avons passés ensemble, je remercie Dieu de t'avoir connue. Je vous souhaite une vie pleine de bonheur.

A personne qui, par un mot, m'ont donné la force de continuer...

Aux tous qui nous aimons, qui étaient toujours à nos côtés, qui nous ont toujours soutenu...

Remerciements

Je souhaiterais tout d'abord, je remercie **ALLAH** le tout Puissant de m'avoir donné la force, la volonté et le courage d'accomplir ce modeste travail.

Ce travail n'aurait pas pu avoir le jour sans l'aide et l'encadrement de Mme **BERGHIDA Meryem**, je tiens à la remercier pour la qualité de son encadrement exceptionnel, pour sa gentillesse, ses conseils, sa patience et sa disponibilité durant la préparation de ce mémoire.

Je tiens à adresser mes vifs et sincères remerciements aux **membres de jury** pour l'honneur qu'ils m'ont fait en acceptant de siéger à notre soutenance.

Mes profonds remerciements vont également à toutes les personnes qui m'ont aidé et soutenu et participé de près ou de loin à la réalisation de ce mémoire. J'espère que vous trouverez à travers ces lignes, les sentiments de ma profonde reconnaissance.

Enfin, je ne peux passer outre ma reconnaissance envers **ma mère, mes sœurs** pour leur patience, aide et leur soutien constant qui me donne des bases solides.

Résumé

Ce travail est la combinaison de plusieurs disciplines qui incluent la mécanique quantique, l'informatique quantique et la génétique. Certains concepts de la mécanique quantique ont été inspirés et associés aux algorithmes génétiques afin d'améliorer leur performance pour résoudre des problèmes d'optimisation combinatoire (dans ce travail, on s'intéresse au problème de sac à dos classique).

De nouveaux opérateurs qu'on a appelé opérateurs génétiques quantiques ont donc été définis à partir des opérateurs génétiques classiques (croisement, mutation) et de concepts quantiques tels que le bit quantique, la superposition d'états et la mesure quantique.

Mots clés : Problème de sac à dos, Optimisation combinatoire, Mécanique quantique, Informatique quantique, Algorithme génétique, Algorithme quantique, Qubit .

Abstract

This work is a combination by multiple disciplinary that include quantum mechanics, quantum computing and genetics. Some concepts of quantum mechanics have been inspired and associated with genetic algorithms in order to improve their performance in solving combinatorial optimization problems (in this work, we are interested in the classic backpack problem).

New operators, which have been called quantum genetic operators, have therefore been defined based on classical genetic operators (crossing, mutation) and quantum concepts such as the quantum bit, the superposition of states and quantum measurement.

Keywords : knapsack problem, Combinatorial optimization, Quantum mechanics, Quantum computing, Genetic algorithm, Quantum algorithm, Qubit.

Table des matières

Introduction générale	1
1 Informatique Quantique (IQ)	3
1.1 Introduction	4
1.2 La mécanique quantique (MC)	4
1.3 Le rayonnement du corps noir	5
1.4 Le chat de Schrödinge	5
1.5 La différence entre la physique classique et la physique quantique	6
1.6 Informatique Quantique (IQ)	6
1.6.1 Le bit quantique (qubit)	7
1.6.2 Deux phénomènes au cœur de l'ordinateur quantique	7
1.6.2.1 La Superposition quantique	7
1.6.2.2 L'intrication quantique	8
1.6.3 La mesure	8
1.6.4 La différence entre l'informatique classique et l'IQ	8
1.7 Domaine d'application de l'IQ	9
1.7.1 IBM au cœur de la révolution de l'IQ	9
1.7.1.1 Apprentissage machine quantique	10
1.7.1.2 Matériaux	10
1.7.1.3 Optimisation quantique	10
1.7.1.4 La finance quantique	10
1.7.1.5 Techniques d'atténuation des erreurs	10
1.7.1.6 Quantum Marketing	11
1.7.1.7 La santé quantique	11
1.7.1.8 Défense et Aérospatial	11

1.7.2	La suprématie quantique de Google	11
1.8	Les algorithmes évolutionnaire inspirés de l'IQ (AEIQ)	12
1.8.1	Structure d'un AEIQ	13
1.8.2	Processus d'évolution	13
1.8.2.1	Codage des individus	13
1.8.2.2	Mesure	13
1.8.2.3	L'interférence	14
1.9	Conclusion	14
2		
	Optimisation Combinatoire	
	Algorithme Génétique Quantique	15
2.1	Introduction	16
2.2	Optimisation Combinatoire	18
2.2.1	Méthodes de résolution	18
2.2.1.1	Méthodes exactes	18
2.2.1.2	Méthodes approchées	19
2.2.2	Complexité d'un problème Optimisation Combinatoire	19
2.2.3	Complexité d'un algorithme	20
2.2.4	Classes de complexités classiques	20
2.3	Algorithme génétique quantique	22
2.3.1	Les Algorithmes génétiques classiques	22
2.3.1.1	Paradigme de la solution	22
2.3.1.2	Le codage binaire	23
2.3.1.3	Les opérateurs de l'algorithme génétique classique	23
2.3.1.3.1	La sélection	23
2.3.1.3.2	Le Croisement	24
2.3.1.3.3	La Mutation	24
2.3.2	Les algorithmes génétiques quantiques	25
2.3.3	Etat de l'art	26
2.4	Conclusion	27
3	Conception	28
3.1	Introduction	29
3.2	La lecture du fichier	29
3.3	Structuration de l'Algorithme	30

3.3.1	Générer la solution initiale aléatoirement	30
3.4	Génération d'une population de solutions	32
3.5	La mesure (Measurement)	33
3.6	Evaluation	34
3.7	MAJ de $Q(t)$ par rotation des portes quantiques pour avoir $Q(t+1)$. . .	35
3.8	Le Croisement	37
3.9	La Mutation	38
3.10	Réctification des solutions infaisables	38
3.11	Amélioration les solutions par l'algorithme de descente	39
3.12	Conclusion	39
4	Résultats et Expérimentation	40
4.1	Introduction	41
4.2	Environnement de développement	41
4.2.1	Environnement Matériel	41
4.2.2	Environnement logiciel	41
4.3	Langages de programmation	42
4.4	Présentation de l'application	43
4.5	Données de tests	46
4.6	Résultats expérimentaux et discussion	47
4.6.1	Comparaison des deux méthodes approchées	47
4.7	Conclusion	56
	Conclusion générale	57
	Bibliographie	58

Table des figures

1	Le chat de Schrödinge	5
2	Le bit quantique.	7
3	Superposition quantique.	8
4	les différences entre le bit classique et le bit quantique.[27]	8
5	La puce Sycamore de Google alimente un ordinateur quantique.[15]	12
6	Structure d'un chromosome quantique.[6]	13
7	Mesure d'un chromosome quantique.[6]	14
1	Le paradigme de la solution.[26]	22
2	Codage binaire d'un chromosome.	23
3	Croisement avec un point[26]	24
4	Croisement avec deux points.[26]	24
5	La mutation.[26]	25
1	Structure d'un AGQ	32
2	Mesurer une solution.[5]	34
3	Lookup pour la rotation des portes quantiques.[23]	36
1	Eclipse IDE	42
2	Java.	43
3	L'interface principale.	43
4	L'interface d'erreur	44
5	choisir «file.txt».. . . .	45
6	L'interface (les champs remplis)	45
7	Instance Benchmark (50 objets).	46

Introduction générale

L'informatique quantique révolutionnera les usages numériques dans les années à venir en démultipliant la puissance de calcul des ordinateurs. Pour accompagner la construction de l'ordinateur quantique et la progression vers les quelques milliers de Qubits intriqués nécessaires à ce saut technologique.

Dans le monde quantique, de nouvelles capacités de calcul permettront le traitement statistique d'énormes quantités de données, en augmentant la rapidité d'analyse de ces derniers.

Une nouvelle approche proposée, appelée Algorithme Evolutionnaire Quantique (QEA), comprend l'utilisation d'un algorithme évolutif à l'aide de l'informatique quantique. En cela, qubit est utilisé comme représentation probabiliste au lieu de représentation binaire, ainsi que l'introduction de portes quantiques pour amener les individus vers la meilleure solution.

Avec l'utilisation de QEA, il a également prouvé sa suprématie sur l'optimisation multi-objectifs, l'optimisation combinatoire et dans la vie réelle, comme la détection des visages, le traitement du signal. etc.

Dans ce mémoire, nous nous intéressons à la résolution des problèmes d'optimisation discrets et comme exemple, nous avons choisi le problème de sac à dos classique (il modélise un sac à dos a une capacité limitée et des objets différents. Pour chaque objet possède une valeur et un poids). Ce problème a été traité à l'aide de combinaison des algorithmes génétiques et l'informatique quantique où les individus manipulés sont des chromosomes quantiques.l'approche proposée a été amélioré en introduisant l'algorithme de descente dans chaque itération.

Ce mémoire comporte quatre chapitres organisés comme suit :

Le premier chapitre présente quelques notions fondamentales liées aux la mécanique quantique, introduction au domaine de l'informatique quantique, l'importance du domaines, les domaines d'application et les algorithmes inspirés de l'informatique quantique en présentant leur structure ainsi que la description détaillée de leur processus d'évolution.

Le deuxième chapitre présente dans la première partie domaine d'optimisation combinatoire, ses méthodes, la complexité et ses classes classiques. Et dans la deuxième partie nous présentons les algorithmes génétiques classiques, Paradigme de la solution, ses opérateurs classiques (croisement, mutation), les algorithmes génétiques quantiques et un état de l'art.

Le troisième chapitre est consacré aux la phase de conception de l'algorithme algorithme génétique quantique pour résoudre le problème du sac à dos, son architecture avec une explication détaillée.

Le quatrième chapitre présente l'implémentation, l'environnement de développement ainsi que le langage de programmation utilisé. Ensuite, les données de tests utilisées, et enfin les résultats obtenus.

Le mémoire est clôturé par une **conclusion générale**.

Chapitre **1**

Informatique Quantique (IQ)

1.1 Introduction

Cela fait déjà plusieurs années que le monde de la science et de la technologie est animé par l'IQ, mais notre quotidien est toujours dépourvu d'une telle technologie.

Les chercheurs ont essayé de créer un ordinateur quantique universel pour un usage commercial au cours des dernières décennies. Les ordinateurs quantiques sont efficaces pour résoudre des problèmes complexes de **classe NP**, car ils peuvent traiter ces problèmes dans un temps polynomial (comme les diagnostics médicaux et les prévisions météorologiques), ce que ne peuvent pas faire les ordinateurs classiques. Ils sont également capables de résoudre efficacement d'énormes problèmes parallèles.

Ce chapitre présente une introduction à l'informatique quantique

Tout d'abord, nous commençons ce chapitre avec un rappel sur la **mécanique quantique** et les trois principes (superposition, l'intrication, la mesure). Nous présentons ensuite **l'informatique quantique** (émergée du domaine de la mécanique quantique) en définissant la notion du qubit, l'état de l'art, les algorithmes inspirés de l'informatique quantique.

1.2 La mécanique quantique (MC)

Développé au début de XXe par les physiciens européens pour résoudre les problèmes que la physique classique comme : le rayonnement du corps noir... etc. La MC est un des piliers de la science contemporaine. Et pourtant, il s'agit aussi probablement de la plus étrange théorie jamais imaginée.[31]

En effet, la MC regorge de mystères, de surprises et de paradoxes qui nous obligent à revoir la manière dont nous concevons la matière, et même la physique en général.[31]

La MC est une domaine de la physique qui explique comment se comportent les particules élémentaires (les électrons, les protons, ...), les atomes ou les molécules, ... c'est-à-dire des objets extrêmement petits dont la taille est inférieure à environ un nanomètre. Les lois de la MC sont très différentes de celles dont nous avons l'habitude. Une particule élémentaire ne se comporte pas du tout comme une petite bille, par exemple, mais de manière bien plus complexe et parfois très surprenante![35]

Par exemple, on ne peut pas savoir où se trouve une particule quantique, on peut seulement trouver un pourcentage de chance qu'un objet se trouve à un endroit précis. On parle alors de probabilité de présence.

Cette théorie est d'ailleurs tellement bizarre que l'un de ses plus fameux contributeurs le physicien Richard Feynman disait à son propos :< **Si vous croyez comprendre la MC, c'est que vous ne la comprenez pas** >.[31]

1.3 Le rayonnement du corps noir

Comme nous le savons, le corps noir est un objet idéal qui absorberait parfaitement toute l'énergie électromagnétique qu'il reçoit, sans en réfléchir ni en transmettre. Il n'est fait aucune autre hypothèse sur la nature de l'objet. Sous l'effet de l'agitation thermique.[9]

En 1900, Planck a émis une hypothèse que l'énergie est émise du corps noir par paquets indivisibles. Suppose aussi que l'énergie transportée par l'onde lumineuse est proportionnelle à sa fréquence (la constante de proportionnalité h).[41]

C'est à dire le rayonnement thermique spontanément émis par les objets ordinaires peut être approximé par un rayonnement de corps noir. Une enceinte parfaitement isolée, en équilibre thermique interne, renferme un rayonnement de corps noir, qui pourra être émis à travers un trou fait dans sa paroi, à condition que le trou soit assez petit pour que son effet soit négligeable sur l'équilibre.[25]

1.4 Le chat de Schrödinger

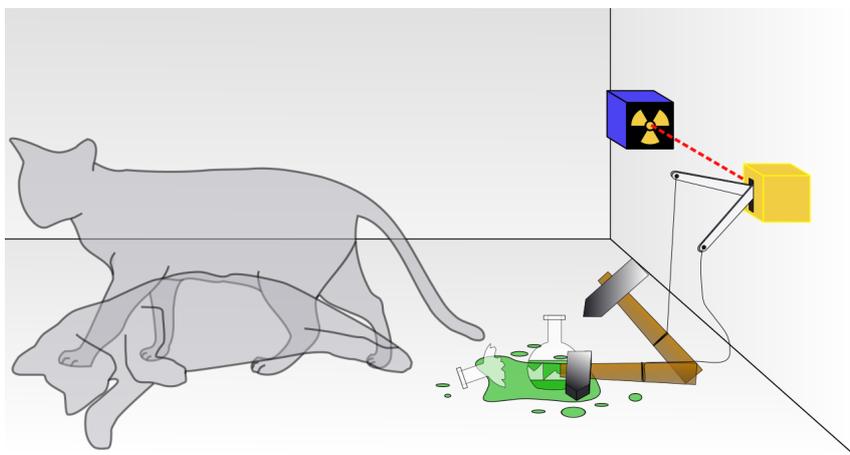


FIGURE 1 – Le chat de Schrödinger

On parler d'étrange superposition à travers l'exemple du fameux chat de Schrödinger, ce chat « fictif » qui serait à la fois mort et vivant. En utilisant cette notation entre crochets, on pourrait écrire : $|Chat \rangle = |Mort \rangle + |Vivant \rangle$.[31]

Bien sûr, l'exemple du chat n'est pas très réaliste, car cette situation ne peut se produire que pour les objets microscopiques.

1.5 La différence entre la physique classique et la physique quantique

Dans la conception classique des lois de probabilités, lorsqu'un événement peut se produire de deux façons différentes incompatibles l'une avec l'autre, les probabilités s'additionnent. Tel n'est pas le cas en MC, où la probabilité d'un évènement est liée à une amplitude de probabilité susceptible d'interférer, y compris de façon destructive.[33] Globalement, la MC se démarque de la physique classique par :

- Niveau de représentation :
 - Physique classique => au niveau macroscopique.
 - Mécanique quantique => au niveau microscopique (les atomes, les particules.)[32]
- Des règles différentes quant à l'additivité des probabilités.
- L'existence de grandeurs physiques ne pouvant se manifester que par multiples de quantités fixes, appelés quanta, qui donnent leur nom à la théorie.[34]

Bien que les scientifiques du siècle dernier, dont Planck et Einstein, aient hésité sur les implications de la théorie quantique, les principes de la théorie ont été étayés à plusieurs reprises par l'expérimentation, même lorsque les scientifiques tentaient de les réfuter. La théorie quantique comme la théorie de la relativité d'Einstein constituent le fondement de la physique moderne. Les principes de la MC s'appliquent à de nombreux domaines, notamment l'optique, la chimie, l'informatique et la cryptographie.[47]

1.6 Informatique Quantique (IQ)

Au début des années 1980, le Nobel de physique Richard Feynman est le premier à pressentir les possibilités faramineuses d'un ordinateur capable de tirer parti des lois quantiques.[30]

Dès les années 1990, plusieurs théoriciens démontrent que certains calculs verraient leur résolution accélérée dans des proportions inouïes s'il était possible de les implémenter sur des bits quantiques. Pour comprendre le fonctionnement d'un ordinateur quantique, pas de mystère vous devez déjà **comprendre les bases de la physique quantique**. [30]

L’IQ est un domaine récent des Sciences et Technologies de l’Information et de la Communication qui est en plein développement. La plupart des grands centres de recherche publics et privés ont créé, ces dernières années, des équipes voire des laboratoires sur ce thème.[49] Est le sous-domaine de l’informatique qui d’étude visant à développer des technologies basées sur les principes de la MC.

Cette théorie explique la nature et le comportement de l’énergie et de la matière au niveau quantique (atomique et subatomique). Contrairement aux ordinateurs d’aujourd’hui qui utilisent le bit comme unité fondamentale, l’ordinateur quantique utilise une valeur appelée le **qubit** (quantum bit).

1.6.1 Le bit quantique (qubit)

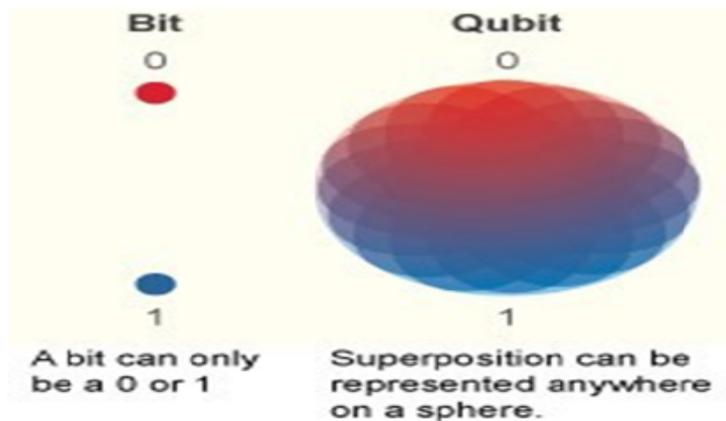


FIGURE 2 – Le bit quantique.

Etat quantique représentant la plus petite unité de stockage d’IQ. Il se compose d’une superposition de deux états de base Ces qubits peuvent donc être à la fois à la valeur 0 et 1.

1.6.2 Deux phénomènes au cœur de l’ordinateur quantique

1.6.2.1 La Superposition quantique

Ce principe dit qu’un qubit peut être prendre plusieurs valeurs au même temps c’est-à-dire si un qubit peut se trouver dans un état $|0\rangle$, et si de même il peut se dans un état $|1\rangle$, alors il peut aussi se trouver dans un état linéairement composé :

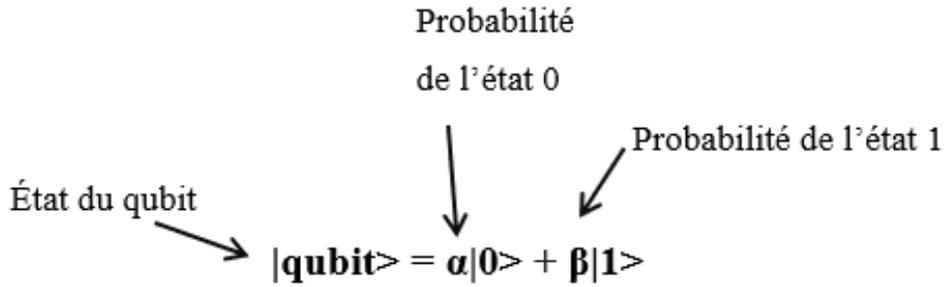


FIGURE 3 – Superposition quantique.

1.6.2.2 L'intrication quantique

Est un phénomène dans lequel deux qubits peuvent également interagir, leurs états s'entremêlant et devenant interdépendants. Par exemple deux objets intriqués O1 et O2 ne sont pas indépendants même séparés, et il faut considérer { O1+O2 } comme un système unique.

1.6.3 La mesure

La mesure quantique est l'action de déterminer la ou les valeurs d'une grandeur (longueur, capacité, etc.), par comparaison avec une grandeur constante de même espèce prise comme terme de référence (étalon ou unité).

1.6.4 La différence entre l'informatique classique et l'IQ

- Si l'informatique **classique** s'appuie sur les **bits**, l'**IQ** repose sur les bits quantiques, ou **qubits**. A la différence de son collègue binaire, le qubit n'est pas cantonné aux seuls 0 et 1, puisqu'il est capable de prendre ces deux valeurs à un même instant. C'est ce qu'on appelle le principe de **superposition** des états. Mais ce n'est pas la seule particularité de ces éléments. Car deux qubits peuvent également interagir, leurs états s'entremêlant et devenant interdépendants. On parle alors d'**intrication**.^[20]
 - par exemple, un état de spin pour un photon ou un électron.

	Bit	
	Classique	Quantique
Élément	Déterministe	Probabiliste
États:	0 ou 1	0 ou 1 ou (0 et 1)
Superposition	Non	Quantique

FIGURE 4 – les différences entre le bit classique et le bit quantique.^[27]

- L'ordinateur quantique, lui, peut fonctionner avec une porte logique à deux modes : XOR et un mode que nous appellerons QO1 (la possibilité de transformer 0 en une su-perposition de 0 et 1, une fonction logique qui n'existe pas en informatique classique).[19]
- Dans un ordinateur quantique, il est possible de faire intervenir un certain nombre de particules élémentaires telles que des électrons ou des photons (en pratique, des ions ont également été utilisés avec succès), leur charge ou leur polarisation agissant comme une représentation de 0 et/ou de 1. Chacune de ces particules est désignée sous le nom de qubit, dont la nature et le comportement forment la base de l'IQ.[19]
- Utiliser des calculs d'un ordinateur classique pour simuler le comportement d'un sys-tème quantique est limité. Avec un ordinateur classique, le temps croît exponentielle-ment avec la complexité du problème. Heureusement, cette croissance n'est que poly-nomiale pour un système quantique. Donc, théoriquement, un ordinateur quantique calcule plus rapidement qu'un ordinateur classique.[1]

1.7 Domaine d'application de l'IQ

Comme toujours, les transformations technologiques s'accompagnent d'opportunités. Ainsi, l'exploitation de la puissance des ordinateurs quantiques peut procurer un avan-tage majeur dans de nombreux domaines, de la santé à la chimie, en passant par l'industrie, la sécurité informatique, les voitures autonomes, l'énergie, ou encore les transports et la gestion du trafic. Les promesses du qubit semblent illimités au point que l'on parle un peu partout d'une "révolution" quantique.

1.7.1 IBM au cœur de la révolution de l'IQ

IBM est un grand fan de l'IQ. **IBM Q** est une première initiative de l'industrie visant à construire des ordinateurs quantiques universels pour les entre-prises, l'ingénierie et la science. Cet effort comprend l'avancement de l'ensemble de la pile de technologies informatiques quantiques et l'exploration d'applications permettant de rendre le quantum utilisable et accessible au sens largetechnologique de l'IQ et l'exploration d'applications pour rendre le quantum largement utilisable et accessible. **IBM Q** accélère l'applica-tion de l'IQ pour résoudre les problèmes les plus com-plexes en chimie, optimisation et apprentissage automatique.[40]

1.7.1.1 Apprentissage machine quantique

L'utilisation de systèmes quantiques pour former et exécuter des algorithmes d'apprentissage machine pourrait nous permettre de résoudre des problèmes complexes plus rapidement, en améliorant potentiellement des applications telles que le diagnostic des maladies, la détection des fraudes et la gestion efficace de l'énergie.[40]

1.7.1.2 Matériaux

La simulation de systèmes quantiques constitue une application précoce prometteuse de l'informatique quantique. Cette technique peut être appliquée à des domaines tels que la chimie, la science des matériaux et la physique des hautes énergies.[40]

1.7.1.3 Optimisation quantique

Les ordinateurs quantiques pourraient potentiellement trouver la meilleure solution parmi différentes options pondérées de manière plus efficace que les ordinateurs classiques, et pourraient offrir un avantage dans des domaines tels que l'optimisation de portefeuille, l'analyse de risque et les applications de type Monte-Carlo.[40]

1.7.1.4 La finance quantique

Dans le secteur des services financiers, il existe de nombreux problèmes informatiques complexes, tels que l'optimisation des portefeuilles financiers ou l'analyse des risques de tels portefeuilles. Pour certains de ces problèmes, l'IQ peut potentiellement offrir un avantage significatif par rapport à l'informatique classique.[40]

La finance est un autre beau terrain de jeu pour expérimenter des algorithmes quantiques. Ce n'est pas par hasard que la première intervention de conférencier sur l'IQ dans une entreprise ait eu lieu **le 5 juillet 2018 à la Société Générale!**. [8]

1.7.1.5 Techniques d'atténuation des erreurs

Travailler à développer des méthodes sur la voie de la correction d'erreur quantique complète qui ont la capacité de réduire considérablement le bruit dans les dispositifs actuels. Bien que l'IQ tolérante aux pannes à grande échelle puisse nécessiter des développements considérables, nous avons développé la technique d'extension quantique en sous-espace afin de faciliter l'utilisation des techniques de correction d'erreur quantique pour améliorer les performances des applications sur des périphériques à court terme.[39]

1.7.1.6 Quantum Marketing

Le marketing est aussi un domaine où les algorithmes d'optimisation de systèmes complexes réalisés à base d'ordinateurs quantiques pourraient être intéressants. Cela concerne l'optimisation du mix marketing, celui de plans médias, ou la maximisation de revenus publicitaires, divers domaines qui sont également investis par le champ de l'IA.[8]

1.7.1.7 La santé quantique

La principale application de l'IQ en santé est la découverte de thérapies via la simulation moléculaire de leur fonctionnement et de celle de leurs cibles, tout en évitant d'éventuelles contre-indications, le tout "in silico". La simulation peut porter sur l'articulation de molécules organiques simples comme le cholestérol ou le repliement des protéines qui est de plusieurs ordres de grandeur plus complexe. Cette dernière prouesse relève donc du très long terme. Elle est aussi à la limite du faisable en termes de complexité car elle est dans la classe des problèmes **NP-Complet**. [8]

1.7.1.8 Défense et Aérospatial

Le complexe militaro-industriel a toujours été un grand consommateur d'informatique de pointe. C'est évidemment le cas aux USA mais aussi en Europe, avec Airbus qui est l'un des premiers industriels à s'intéresser aux applications du quantique.

En 2015, Raytheon et IBM démontraient l'efficacité d'un algorithme quantique utilisant une "boîte noire" pour reconstruire une chaîne de bits inconnue, le tout fonctionnant sur un ordinateur quantique universel d'IBM de 5 qubits.

Le groupe Airbus a de son côté créé une équipe basée sur leur site de Newport au Pays de Galle, qui s'attaque aux usages du quantique, notamment dans l'analyse d'imagerie aérienne ou pour la conception de nouveaux matériaux.[8]

1.7.2 La suprématie quantique de Google

Google a confirmé le **23 octobre 2019** avoir réalisé une percée majeure dans le domaine de la recherche en IQ. Le géant californien a mis au point une machine capable de résoudre en quelques minutes seulement un problème qui prendrait des milliers d'années aux ordinateurs les plus puissants actuellement.[14]

Un ordinateur quantique de **Google** a largement dépassé la technologie informatique ordinaire, une réalisation appelée suprématie quantique qui constitue un jalon important pour une méthode de traitement des données révolutionnaire. Cette réalisation est le fruit

de plus d'une décennie de travaux chez **Google**, notamment de l'utilisation de sa propre puce IQ, appelée **Sycamore**. [15]

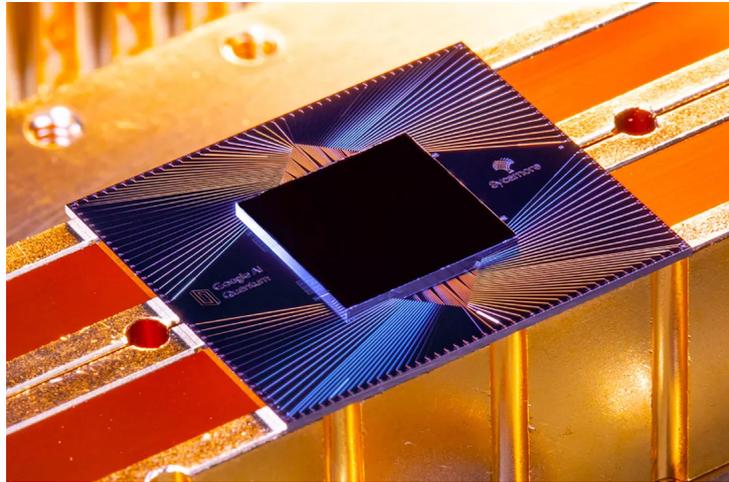


FIGURE 5 – La puce Sycamore de Google alimente un ordinateur quantique. [15]

Selon **Google**, son processeur **Sycamore 54-qubit** n'a besoin que de **3 minutes et 20 secondes** pour effectuer un calcul que les traditionnels superordinateurs auraient mis 10 000 ans à faire. [14]

Les premiers clients de **Google** - le département américain de l'Énergie et les constructeurs automobiles Daimler et Volkswagen - pourront utiliser la machine en **2020**, a annoncé **Google**. Comme dans le cas de l'effort d'IQ d'**IBM**, il sera disponible en tant que service d'informatique en cloud sur Internet. [15]

1.8 Les algorithmes évolutionnaire inspirés de l'IQ (AEIQ)

Une nouvelle approche proposée, appelée Algorithme évolutif inspiré de Quantum, comprend l'utilisation d'un algorithme évolutif à l'aide de l'IQ. En cela, qubit est utilisé comme représentation probabiliste au lieu de représentation binaire ou symbolique, ainsi que l'introduction de portes quantiques pour amener les individus vers la meilleure solution. Cela ne converge pas prématurément dans le cas de population petite et même nombreuse. Avec l'utilisation de QEA [42], il a également prouvé sa suprématie sur l'optimisation multi-objectifs, l'optimisation combinatoire et dans la vie réelle, comme la détection des visages, le traitement du signal. etc. [42]

Un algorithme évolutif polyvalent inspiré de Quantum a été introduit en 2007 avec l'explication du ralentissement de la découverte de la solution optimale dans QEA. Le

remplacement des attracteurs s'effectue sans prendre en compte leur aptitude.[42]

1.8.1 Structure d'un AEIQ

Un AEIQ est un algorithme probabiliste, similaire aux autres AE, qui fait évoluer une population d'individus appelés chromosomes quantiques. Différentes représentations peuvent être utilisées pour coder les individus d'un AE ordinaire. Ces représentations peuvent être binaire, numérique ou encore autre. L'AEIQ utilise une nouvelle représentation probabiliste basée sur le concept de qubit. Un chromosome quantique n'est rien d'autre qu'une chaîne de bits quantiques (qubit, la plus petite unité d'information en IQ) représentés par un couple de nombre (α, β) où :

$$|\alpha|^2 + |\beta|^2 = 1.$$

$|\alpha|^2$ est la probabilité que le qubit soit trouvé dans l'état 0 alors que $|\beta|^2$ est la probabilité que le qubit soit trouvé dans l'état 1. Le qubit peut donc être 0, 1 ou dans une superposition des deux états. Chaque individu est donc représenté par une superposition d'états de toutes les solutions possibles du problème à résoudre.[6]

1.8.2 Processus d'évolution

1.8.2.1 Codage des individus

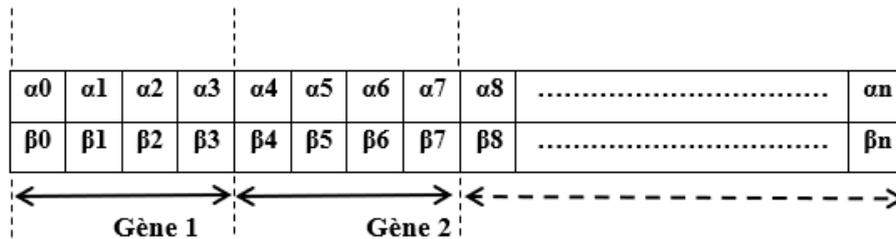


FIGURE 6 – Structure d'un chromosome quantique.[6]

1.8.2.2 Mesure

Cette opération conduit à extraire un chromosome classique à partir d'un chromosome quantique et ceci en mesurant les états un par un.

Pour chaque Qubit, nous générons un nombre aléatoire $r \in [0; 1]$: Si $(r > \alpha^2)$, nous met-tons le bit à 1 sinon nous le mettons à 0.[6]

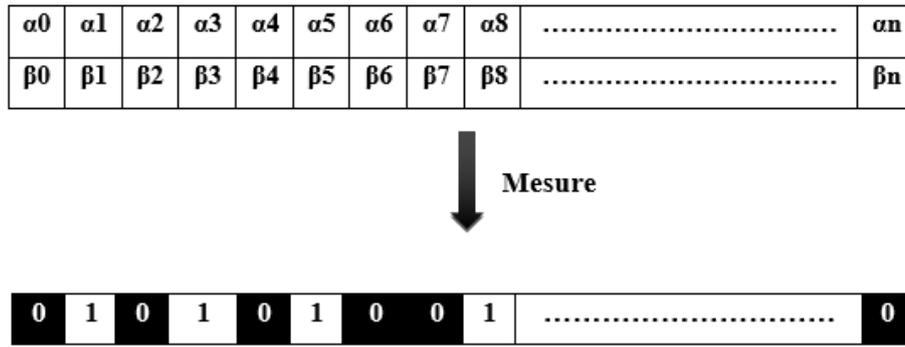


FIGURE 7 – Mesure d’un chromosome quantique.[6]

1.8.2.3 L’interférence

L’interférence est un opérateur d’évolution qui guide les individus vers de meilleures solutions, et finalement, vers un seul état.

Elle peut être réalisée à l’aide d’une transformation unitaire qui permet une rotation dont l’angle est une fonction des amplitudes (α_i, β_i) et de la valeur du bit correspondant à la solution référence. La valeur de l’angle de rotation doit être choisie de manière à éviter la convergence prématurée.[6]

1.9 Conclusion

L’élément humain est au cœur des activités et le restera, mais les capacités d’assistance que l’intelligence artificielle et l’IQ offriront à collaborateurs, sont un avantage significatif dans une industrie où la concurrence est toujours plus vive.

Il est difficile de définir un avenir de l’IQ car C’est une technologie très prometteuse.

En effet, un AEIQ peut explorer un espace de recherche avec un nombre très réduit d’opérations et donc exploite cet espace dans un temps raisonnable.

Chapitre **2**

Optimisation Combinatoire
Algorithme Génétique Quantique

2.1 Introduction

Plusieurs problèmes issus de domaines divers se ramènent à maximiser (ou minimiser) une fonction linéaire sous des contraintes linéaires avec des variables bivalentes. Ces problèmes, dits d'optimisation combinatoire, sont généralement **NP-difficiles**. Des méthodes efficaces ont été ainsi développées pour formuler et résoudre ce type de problèmes.

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. 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.[17]

L'Optimisation combinatoire définit un cadre formel pour de nombreux problèmes de l'industrie, de la finance ou de la vie quotidienne. Etant donnée l'importance de ces problèmes, de nombreuses méthodes de résolution a été développées en recherche opérationnelle (RO) et en intelligence artificielle (IA).

Parmi les méthodes de résolution proposées, on trouve les métaheuristiques. Ces dernières peuvent être réparties dans deux classes : les métaheuristiques à solution unique et les métaheuristiques à population (ensemble de solutions) tel que les algorithmes évolutionnaires (les algorithmes génétiques) que nous avons combiné avec l' informatique quantique pour réaliser notre objectif (résoudre le problème du sac à dos).

Vu que les algorithmes quantiques purs sont difficiles à concevoir en raison de la non disponibilité actuellement des machines quantiques. Plusieurs chercheurs s'intéressent à la combinaison des algorithmes classiques aux principes de l'informatique quantique. L'avantage des algorithmes inspirés de la mécanique quantique est qu'ils ne nécessitent pas la présence d'ordinateurs quantiques. Le premier algorithme inspiré du quantique est celui de Moore pour le tri de nombres. En 2000, Han et Kim ont proposé le premier algorithme génétique inspiré du quantique. En 2001, ils proposent un algorithme quantique évolutionnaire.[46] Cet algorithme combine les idées de l'informatique quantique aux algorithmes évolutionnaires classiques. Les algorithmes inspirés de la mécanique quantique ont donné de bons résultats face à des problèmes d'optimisation combinatoire comme le problème du recalage d'image, le problème de sac à dos, le problème d'allocation de disque etc.

Depuis la fin des années 1990, l'hybridation du calcul quantique et des algorithmes évolutionnaires s'est avérée être une solution efficace pour traiter des problèmes complexes. Dans ce contexte, l'algorithme génétique quantique (AGQ) a prouvé son efficacité pour la résolution de problèmes d'optimisation. AGQ peut assurer l'équilibre entre exploration et exploitation plus facilement que l'algorithme génétique classique. D'autre part, il peut explorer l'espace de recherche avec un nombre d'individus réduit pour obtenir une solution globale en un temps minimal.

AGQ est aussi caractérisé par la représentation des individus, la taille de la population, la fonction de fitness et la dynamique de la population.

La première partie dans ce chapitre englobe quelques notions élémentaires de la théorie de l'optimisation combinatoire. Dans un premier temps on va commencer la définition de l'optimisation combinatoire et Méthodes de résolution. Ensuite, la complexité d'un problème et d'un algorithme. Ensuite on va définir quelques classes de problèmes d'optimisation

Dans la deuxième partie dans ce chapitre on va aborder les algorithmes génétiques et les opérations génétiques (Croisement, mutation). Ensuite, la relation entre les algorithmes génétiques et les algorithmes génétiques quantique. Enfin, on va présenter quelques travaux connexes.

2.2 Optimisation Combinatoire

L'optimisation combinatoire on dit aussi optimisation discrète est une branche de l'optimisation en mathématiques appliquées et en informatique, également liée à la recherche opérationnelle, l'algorithmique et la théorie de la complexité.

L'Optimisation combinatoire est le domaine des mathématiques discrètes qui consiste à trouver un "meilleur" solution parmi un ensemble fini (souvent très grand) de possibilités. Pour définir la notion de meilleure solution, une fonction, dite fonction objectif (ou coût), est introduite. Pour chaque solution, elle renvoie un réel et la meilleure solution (ou solution optimale) est celle qui minimise ou maximise la fonction objectif. Clairement, un problème d'optimisation combinatoire peut avoir plusieurs solutions optimales.[36]

L'optimisation combinatoire trouve des applications dans des domaines aussi variés que la gestion, l'ingénierie, la conception, la production, les télécommunications, les transports, l'énergie, les sciences sociales et l'informatique elle-même.

Un problème d'optimisation combinatoire (CO) est défini par un ensemble d'instances. A chaque instance du problème est associé un ensemble discret de solutions S , un sous-ensemble X de S représentant les solutions admissibles (réalisables) et une fonction objectif f qui assigne à chaque solution $s \in X$ le nombre réel (ou entier) $f(s)$. Résoudre un tel problème (plus précisément une telle instance du problème) consiste à trouver une solution $s^* \in X$ optimisant la valeur de la fonction de coût f . Une telle solution s^* s'appelle une solution optimale ou un optimum global.[17]

Un problème est dit polynomial s'il existe un algorithme permettant de trouver une solution optimale pour toutes ses instances en un temps polynomial par rapport à la taille de l'instance. Un tel algorithme est dit efficace pour le problème en question. Pour la majorité des problèmes d'optimisation combinatoire, aucun algorithme polynomial n'est connu actuellement.[17]

2.2.1 Méthodes de résolution

2.2.1.1 Méthodes exactes

Les méthodes de résolution exactes sont nombreuses et se caractérisent par le fait qu'elles permettent d'obtenir une ou plusieurs solutions dont l'optimalité est garantie.

Parmi ces méthodes, on peut remarquer l'algorithme du simplexe qui permet d'obtenir la solution optimale d'un problème en parcourant la fermeture convexe de l'ensemble de recherche (ensemble des solutions admissibles) et ce en passant de sommet en sommet.

Malgré une complexité mathématique dans le pire des cas non polynomiale, il permet de résoudre la plupart des problèmes rapidement.[38]

2.2.1.2 Méthodes approchées

Dans certaines situations, il est nécessaire de disposer d'une solution de bonne qualité (c'est-à-dire assez proche de l'optimale) dans un contexte de ressources (temps de calcul et/ou mémoire) limitées. Dans ce cas l'optimalité de la solution ne sera pas garantie, ni même l'écart avec la valeur optimale. Cependant, le temps nécessaire pour obtenir cette solution sera beaucoup plus faible et pourra même être fixé (bien évidemment dans ce cas la qualité de la solution obtenue dépendra fortement du temps laissé à l'algorithme pour l'obtenir).[38]

Typiquement ce type de méthodes, dites heuristiques¹ est particulièrement utile pour les problèmes nécessitant une solution en temps réel (ou très court) ou pour résoudre des problèmes difficiles sur des instances numériques de grande taille. Elles peuvent aussi être utilisées afin d'initialiser une méthode exacte.[38]

2.2.2 Complexité d'un problème Optimisation Combinatoire

La théorie de la complexité vise à savoir si la réponse à un problème peut être donnée très efficacement, efficacement ou au contraire être inatteignable en pratique, avec des niveaux intermédiaires de difficulté entre les deux extrêmes ; pour cela, elle se fonde sur une estimation – théorique – des temps de calcul et des besoins en mémoire informatique. Dans le but de mieux comprendre comment les problèmes se placent les uns par rapport aux autres, la théorie de la complexité établit des hiérarchies de difficulté entre les problèmes algorithmiques, dont les niveaux sont appelés des « classes de complexité ». Ces hiérarchies comportent des ramifications, suivant que l'on considère des calculs déterministes – l'état suivant du calcul est « déterminé » par l'état courant – ou non déterministes.

La complexité d'un problème est la complexité du meilleur algorithme qui permet de le résoudre. Si cet algorithme est polynomial, le problème est dit facile, autrement le problème est difficile. Nous présentons ici la complexité en temps. Il existe aussi une mesure de la performance des algorithmes en termes d'espace mémoire dite complexité en espace.[10]

Complexité d'un problème est une estimation du nombre d'instructions à exécuter pour résoudre les instances de ce problème, cette estimation étant un ordre de grandeur par rapport à la taille de l'instance. Le sens où la complexité d'un problème est définie en considérant son instance la plus difficile. Les travaux théoriques dans ce domaine ont permis d'identifier différentes classes de problèmes en fonction de la complexité.[43]

- La classe P contient l'ensemble des problèmes polynomiaux, pouvant être résolus par un algorithme de complexité polynomiale. Cette classe caractérise l'ensemble des problèmes que l'on peut résoudre « efficacement ».[43]
- La classe NP contient l'ensemble des problèmes polynomiaux non déterministes. Ces problèmes sont liés par une relation d'équivalence, pouvant être résolus par un algorithme de complexité polynomiale pour une machine non déterministe (que l'on peut voir comme une machine capable d'exécuter en parallèle un nombre fini d'alternatives). Tous les problèmes **NP-difficiles** pourront être résolus en un temps polynomial. Cette classe englobe les problèmes les plus.[43]

2.2.3 Complexité d'un algorithme

La complexité d'un algorithme va naturellement être fonction de la taille des données passées en paramètres. Cette dépendance est logique, plus ces données seront volumineuses, plus il faudra d'opérations élémentaires pour les traiter.[13]

Souvent la complexité dépend aussi de la donnée en elle-même et pas seulement de sa taille. En particulier la façon dont sont réparties les différentes valeurs qui la constituent. On devra en effet distinguer trois formes de complexité en temps :

- **La complexité dans le meilleur des cas** : c'est la **situation la plus favorable**, qui correspond par exemple à la recherche d'un élément situé à la première position d'une liste, ou encore au tri d'une liste déjà triée.
- **La complexité dans le pire des cas** : c'est la **situation la plus défavorable**, qui correspond par exemple à la recherche d'un élément dans une liste alors qu'il n'y figure pas, ou encore au tri par ordre croissant d'une liste triée par ordre décroissant.
- **La complexité en moyenne** : on suppose là que les données sont réparties selon une certaine loi de probabilités.[13]

L'objectif premier d'un calcul de complexité algorithmique est de pouvoir comparer l'efficacité d'algorithmes résolvant le même problème. Dans une situation donnée, cela permet donc d'établir lequel des algorithmes disponibles est le plus optimal.[13]

2.2.4 Classes de complexités classiques

- $O(\log n)$ Ce sont des algorithmes très rapides. Exemples typiques : recherche dichotomique, exponentiation rapide, etc.

- $O(n)$ (on dit linéaire). Typiquement quand on parcourt un tableau ou une liste un nombre borné de fois : recherche dans un tableau, minimum d'une liste, etc.
- $O(n \log n)$. Vous l'avez principalement vu pour les algorithmes efficaces de tri : tri rapide, tri fusion, tri par tas, etc. Cette complexité apparait régulièrement lorsque l'on fait du "diviser pour régner".
- $O(n^2)$ (on dit quadratique). Quand on manipule des tableaux à deux dimensions, ou qu'on effectue un assez grand nombre de calculs sur un tableau à une dimension : somme de deux matrices, transposée d'une matrice, tri insertion, tri bulle, tri selection, etc.[7]

2.3 Algorithme génétique quantique

2.3.1 Les Algorithmes génétiques classiques

Les algorithmes génétiques classique sont des méthodes basées sur les méca-nismes bio-logiques tels que les lois de Mendel et sur le principe fondamental (sélection) de Charles Darwin [1859]. Holland exposa les principes de ces algorithmes pour per-mettre aux ordi-nateurs ”d’imiter les êtres vivants en évoluant” pour rechercher la solu-tion à un problème. Il expliqua d’abord comment ajouter de l’intelligence dans un pro-gramme informatique avec les croisements (échange du matériel génétique) et la muta-tion (source de la diver-sité génétique).[22]

Plus récemment, Goldberg enrichit la théorie des algorithmes génétiques en s’appuyant sur le parallèle suivant [1989] :

- un individu est lié à un environnement par son code d’ADN,
- une solution est liée à un problème par son indice de qualité,
- une ”bonne” solution à un problème donné peut être vue comme un individu susceptible de survivre dans un environnement donné.[22]

2.3.1.1 Paradigme de la solution

Un chromosome est une suite de gène, on peut par exemple choisir de regrouper les paramètres similaires dans un même chromosome (chromosome à un seul brin) et chaque gène sera repérable par sa position : son locus sur le chromosome en question.

Chaque individu est représenté par un ensemble de chromosomes, et une population est un ensemble d’individus.

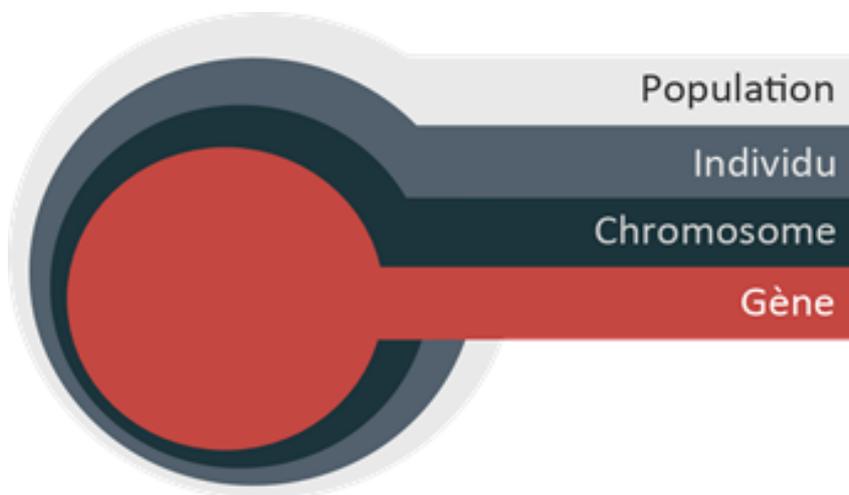


FIGURE 1 – Le paradigme de la solution.[26]

2.3.1.2 Le codage binaire

C'est le plus utilisé. Le codage binaire consiste à utiliser des bits (0 ou 1) pour représenter les différentes solutions. Dans le sens où, les gènes sont représentés par des bits et les chromosomes sont représentés par des chaînes de bits. Le type de ce codage s'adapte bien aux problèmes de type binaire, comme les problèmes du sac à dos. Prenons l'exemple du problème du sac à dos dans sa forme la plus simple (unidimensionnelle). Supposant, que nous devons sélectionner un sous ensemble d'objets parmi un ensemble de 10 objets. Les solutions peuvent être facilement représentées par des chaînes binaires de taille 10, où chaque bit représente l'état d'un objet : le bit 1 pour dire que l'objet est sélectionné, le bit 0 pour dire que l'objet est écarté.[12]

0	1	1	0	1	0	0	1	1	1
---	---	---	---	---	---	---	---	---	---

FIGURE 2 – Codage binaire d'un chromosome.

2.3.1.3 Les opérateurs de l'algorithme génétique classique

2.3.1.3.1 La sélection

La sélection consiste à choisir les individus les mieux adaptés afin d'avoir une population de solution la plus proche qui converge vers l'optimum global.

Cet opérateur est peut-être le plus important puisqu'il permet aux individus d'une population de survivre, de se reproduire ou de mourir. En règle générale, la probabilité de survie d'un individu sera directement liée à son efficacité relative au sein de la population.[26]

Il existe plusieurs techniques de sélection. Voici les principes utilisés : [26]

- **Sélection par rang** : Cette technique de sélection choisit toujours les individus possédant les meilleurs scores d'adaptation.
- **Probabilité de sélection proportionnelle à l'adaptation** : Technique de la roulette ou roue de la fortune, pour chaque individu, la probabilité d'être sélectionné est proportionnelle à son adaptation au problème.
- **Sélection par tournoi** : Cette technique utilise la sélection proportionnelle sur des paires d'individus, puis choisit parmi ces paires l'individu qui a le meilleur score d'adaptation.
- **Sélection uniforme** : La sélection se fait aléatoirement, uniformément et sans intervention de la valeur d'adaptation.

2.3.1.3.2 Le Croisement

Le croisement consiste à générer un ou deux nouveaux individus à partir d'un couple de parents choisi par l'opérateur de sélection. C'est un opérateur qui permet la création de nouveaux individus en combinant les gènes des individus parents.[26]

Il existe deux méthodes de croisement : simple ou double croisement.

- Le croisement simple consiste à fusionner les particularités de deux individus à partir d'un pivot, afin d'obtenir un ou deux enfants :

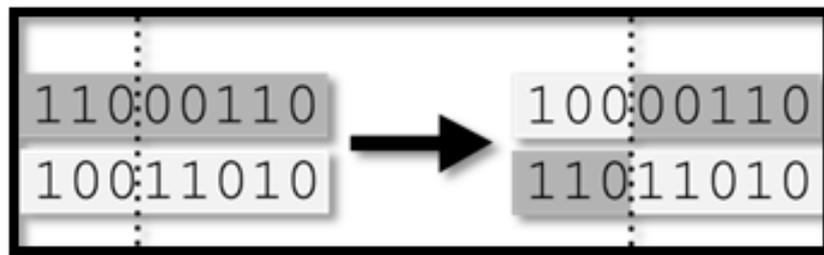


FIGURE 3 – Croisement avec un point[26]

- Le croisement double repose sur le même principe, sauf qu'il y a deux pivots :

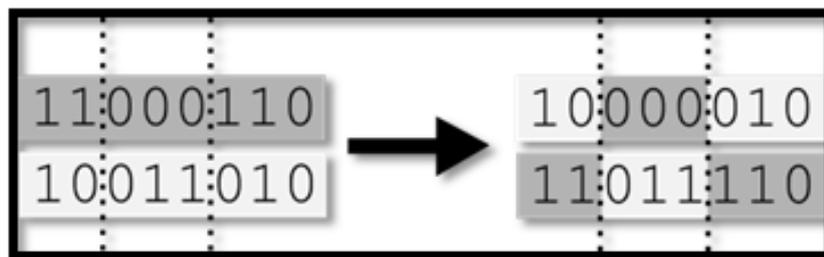


FIGURE 4 – Croisement avec deux points.[26]

2.3.1.3.3 La Mutation

Une mutation consiste à altérer un gène dans un chromosome selon un facteur de mutation Ou 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.[26]

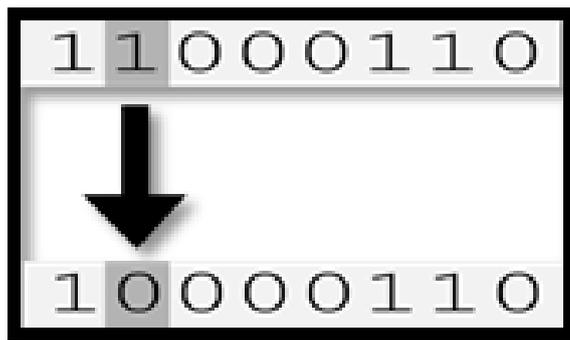


FIGURE 5 – La mutation.[26]

2.3.2 Les algorithmes génétiques quantiques

Un algorithme quantique est une succession d'opération quantique les algorithmes quantiques offrent une complexité nettement inférieure grâce au concept de superposition d'états offert par les principes quantiques.[2]

Un algorithme génétique quantique (AGQ) est un algorithme génétique où les individus manipulés sont des chromosomes quantiques. La représentation des chromosomes est basée sur le concept de qubit, et dotée d'opérations quantiques. Cela induit que les opérations génétiques classiques (croisement, mutation) seront totalement redéfinies pour s'adapter à la nouvelle représentation des chromosomes.[11]

Les algorithmes génétiques quantiques (AGQ) sont la combinaison des AGs et le l'informatique quantique. Il y a eu de grands efforts pour utiliser les AGQ pour explorer des espaces de recherche, nous citons par exemple qui a proposé un AG qui inspire des principes de la mécanique quantique en le testant sur le problème type du sac à dos, a proposé une version parallèle des AGQ pour résoudre les problèmes discrets .[24]

Les AGQ sont une combinaison entre les AG et le calcul quan-tique. Ils sont principalement basés sur les qu-bits et la superposition d'états de la mécanique quantique. Contrairement à la représentation classique des chromosomes (chaines binaires par exemple), ici les chromosomes sont représentés par des vecteurs de qubits (registres quantiques). Ainsi, un chromosome peut représenter la superposition de tous les états possibles.

Les AGQ sont principalement basés sur les qubits et la superposition d'états de la mécanique quantique. Contrairement à la représentation classique des chromosomes (chaines binaires par exemple), ici les chromosomes sont représentés par des vecteurs de qubits (registres quantiques). Ainsi, un chromosome peut représenter la superposition de tous les états possibles. Comparé à l'AG conventionnel, le QGA se caractérise par une

petite taille de population, une convergence rapide et une forte capacité de recherche mondiale.[24]

2.3.3 Etat de l'art

En **2012**, les auteurs [3] ont proposé un algorithme génétique quantique pour résoudre problème segmentation d'image qui est une tâche de traitement de bas niveau qui consiste à partitionner une image en régions homogènes. Le but est de partitionner une image en régions homogènes selon un critère spécifique. Le critère peut être l'intensité lumineuse, la texture, la couleur. La segmentation peut être considérée comme un problème d'optimisation combinatoire. En **2007**, les auteurs [44] ont proposé un algorithme de contrôle de la topologie basé sur l'algorithme de génétique quantique dans les réseaux de capteurs en objectif d'une faible consommation d'énergie et de connectivité élevée car le contrôle de la topologie est crucial. En **2009**, les auteurs [50] ont proposé un algorithme génétique quantique amélioré pour résoudre problème d'horaire des cours, a été prouvé être NP-Complete. En raison de sa complexité, par conséquent, il est très difficile à résoudre en utilisant des techniques d'optimisation conventionnelles.

En **2010**, les auteurs [45] ont proposé un algorithme génétique quantique pour la planification de trajectoires de robots mobiles en utilisant le bit quantique avec l'état de superposition pour coder une route et optimisation en introduisant des opérateurs génétiques. En **2012**, les auteurs [28] ont proposé un algorithme génétique quantique amélioré pour la planification des ateliers, ils suggèrent de présenter les codes sous forme de chromosome quantique afin de résoudre le problème de programmation de l'atelier. La planification des tâches a pour but d'affecter les pièces à chaque machine selon une certaine séquence et d'accomplir le processus de travail avec le minimum de temps.

En **2016**, les auteurs [29] ont proposé un algorithme génétique quantique afin d'accomplir une tâche spécifique dans un système multi-robots, les robots doivent travailler ensemble à travers la formation d'une coalition. la cible de la coalition de robots espère obtenir le maximum d'avantages à un coût minimum qui consiste à trouver la coalition de robots optimale. En **2018**, les auteurs [51] ont proposé un algorithme génétique quantique pour résoudre le problème l'identification de l'authenticité des données d'enregistrement numérique du locuteur, en tant que base importante de l'identification judiciaire.

En **2019**, les auteurs [48] ont proposé un algorithme génétique quantique amélioré (IQGA) : une méthode pour résoudre le problème de la congestion du trafic urbain dans un environnement de Big Data à travers la sélection des itinéraires, Les recherches existantes se sont principalement concentrées sur les trois points suivants : algorithmes de contrôle de la période des feux de signalisation aux intersections simples ; algorithmes de recommanda-

tion d'itinéraire pour un seul véhicule ; et des algorithmes de recommandation d'itinéraire basés sur l'historique du trafic vécu par un véhicule.

2.4 Conclusion

L'optimisation combinatoire est un vaste domaine qui fait intervenir plusieurs types de compétences Comprendre le schéma général pour attaquer un problème : Analyse, Modélisation, Résolution, Interpretation, Application Comprendre les avantages et limitations des différentes méthodes afin de choisir la plus adaptée.

L'absence d'un algorithme déterministe capable de résoudre, de façon optimale et dans un temps de calcul raisonnable, les problèmes d'optimisation combinatoire (POCs) **NP-difficiles**, est le motif de s'orienter vers des algorithmes approximatifs de recherche en l'occurrence les métaheuristiques. Ce choix est contraint par une obligation d'adapter ces métaheuristiques à une recherche dans les espaces combinatoires.[37]

Chapitre **3**

Conception

3.1 Introduction

Dans ce chapitre, nous abordons le problème du sac à dos est un problème d'optimisation combinatoire. Il modélise un sac à dos a une capacité limitée et des objets. Pour chaque objet possède une valeur et un poids.

Notre objectif est de maximiser les coûts des objets mis dans le sac en utilisant les algorithmes génétiques Quantiques.

Les algorithmes génétiques quantiques (AGQ) s'inspirent de la combinaison des algorithmes génétiques et l'informatique quantique où les individus manipulés sont des chromosomes quantiques, nous allons étudier des cas pratiques, en expliquant leur fonctionnement et les implémenter à l'aide du langage JAVA.

3.2 La lecture du fichier

Dans cette étape, nous lisons les données à partir d'un fichier[4]. Puis nous introduisons ces données dans les structures de données, Cette procédure est implémentée dans la méthode **ReadInstance()**.

ArrayList <Objet> Objets : une liste qui contient les objets, de type objet.

ArrayList <Integer> Poids : une liste qui contient les poids des objets, de type entier.

ArrayList <Integer> Valeurs : cette liste est de type entier et qui contient les valeurs des objets.

Structure du fichier

C	Capacité du sac.
n	Nombre des objets.
P1 V1	
P2 V2	
.. ..	Pi : Poids le i -ème objet $i = \{ 1, 2, \dots, n \}$.
.. ..	Vi : Valeur le i -ème objet $i = \{ 1, 2, \dots, n \}$.
.. ..	
.. ..	
Pn Vn	

3.3 Structuration de l'Algorithme

Nous proposons d'utiliser un algorithme inspiré de l'informatique quantique pour résoudre le problème du sac à dos. Algorithmes d'inspiration quantique (QIA) sont de nouveaux algorithmes opérant sur les machines conventionnelles et enrichis par les concepts de l'informatique quantique. La particularité d'algorithme génétique quantique vient de la représentation quantique qu'ils adoptent et dotée d'opérations quantiques. Cela induit que les opérations génétiques classiques (croisement, mutation) seront totalement redéfinies pour s'adapter à la nouvelle représentation des chromosomes, qui permet de représenter la superposition de toutes les solutions possibles pour un problème donné.

- Le Q-bit représentant la plus petite unité de stockage d'information quantique.
- Le pseudo code de notre approche proposée AGQA est illustré dans l'algorithme 1.

Algorithme 1

1. Génération d'une solution initiale aléatoirement.
2. Génération d'une population initiale.
3. La mesure.
4. Récitification des solutions infaisables.
5. Evaluation.
6. Pour $i=1$ à nbre itération Faire
 - L'interférence.
 - Le Croisement.
 - La Mutation.
 - La mesure.
 - Récitification des solutions infaisables.
 - Amélioration de la meilleure solution trouvé S^* .
7. Fin pour
Retourner S^* .
8. Fin

3.3.1 Générer la solution initiale aléatoirement

La solution est représentée par un vecteur de Q-bits de taille n , tel que n est le nombre des objets.

Le Qubit est représenté par une superposition α et β , où :

α : Probabilité de l'état 0. C'est à dire l'objet n'est pas dans le sac.
 β : Probabilité de l'état 1. C'est à dire l'objet est dans le sac

Le vecteur de Qubits exprimé comme suit :

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & .. & \alpha_n \\ \beta_1 & \beta_2 & \beta_3 & .. & \beta_n \end{bmatrix}$$

α_i : Probabilité de l'état 0 de i -ème Q-bit, $i = \{ 1, 2, \dots, n \}$

β_i : Probabilité de l'état 1 de i -ème Q-bit, $i = \{ 1, 2, \dots, n \}$

Algorithme 2

Générer un vecteur vide **Qubit**

Pour $i = 0$ à n **Faire**

Générer un nombre aléatoire $\alpha \in [0, 1]$

Calculer un nombre β **par soustraction** $1 - \alpha$

Nous ajoutons α **et** β **au Qubit**

Fin

Exemple

supposons que nous avons un sac de capacité Q et 15 objet. une solution est présentée comme un vecteur de Q-bit de taille 15. Chaque élément de ce vecteur contient deux valeur α (probabilité que l'objet ne soit pas dans le sac) et β (probabilité que l'objet soit dans le sac).

0.90	0.65	0.50	0	0.80	0.30	0.44	0.18	0.99	0.51	0	0.22	1	0.05	0.88
0.10	0.35	0.50	1	0.20	0.70	0.56	0.82	0.01	0.49	1	0.78	0	0.95	0.12

$$\alpha = 0.90, 0.65, 0.50, 0, 0.80, \dots$$

$$\beta = 0.10, 0.35, 0.50, 1, 0.20, \dots$$

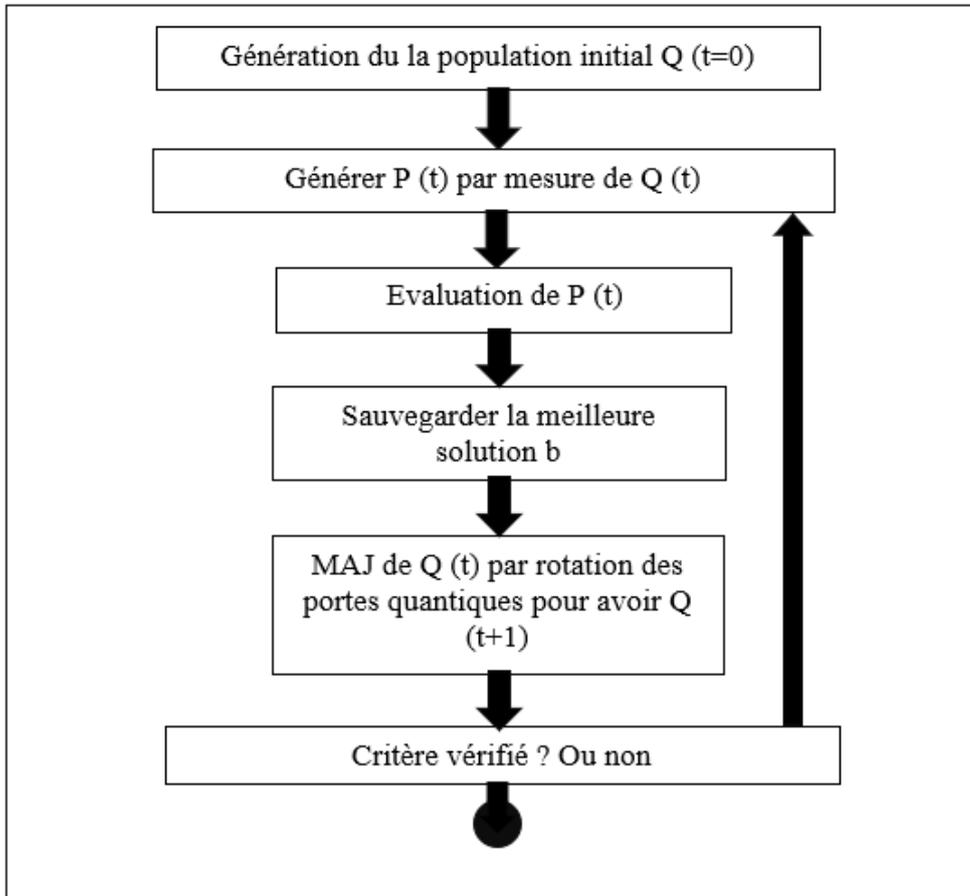


FIGURE 1 – Structure d'un AGQ

3.4 Génération d'une population de solutions

Dans la première étape, un ensemble de solutions sont générées de manière aléatoire et stockées dans POPULATION, Chaque solution est un vecteur de Q-bit.

POPULATION est considérée comme une liste contenant un ensemble de vecteurs représentant des solutions, comme suit :

$$\text{POPULATION} = \left(\begin{array}{c} \mathbf{S0} \left[\begin{array}{ccccccc} \alpha_1 & \alpha_2 & \alpha_3 & \dots & \dots & \dots & \alpha_n \\ \beta_1 & \beta_2 & \beta_3 & \dots & \dots & \dots & \beta_n \end{array} \right] \\ \mathbf{S1} \left[\begin{array}{ccccccc} \alpha_1 & \alpha_2 & \alpha_3 & \dots & \dots & \dots & \alpha_n \\ \beta_1 & \beta_2 & \beta_3 & \dots & \dots & \dots & \beta_n \end{array} \right] \\ \mathbf{S2} \left[\begin{array}{ccccccc} \alpha_1 & \alpha_2 & \alpha_3 & \dots & \dots & \dots & \alpha_n \\ \beta_1 & \beta_2 & \beta_3 & \dots & \dots & \dots & \beta_n \end{array} \right] \\ \dots \\ \mathbf{Sn} \left[\begin{array}{ccccccc} \alpha_1 & \alpha_2 & \alpha_3 & \dots & \dots & \dots & \alpha_n \\ \beta_1 & \beta_2 & \beta_3 & \dots & \dots & \dots & \beta_n \end{array} \right] \end{array} \right)$$

Algorithme 3

Pour $i = 0$ à taille de population Faire
 Générer une solution aléatoire (vecteur de Q-bit).
 Nous ajoutons la solution à la liste de population « POPULATION »
 Fin

3.5 La mesure (Measurement)

La mesure consiste à transformer un vecteur quantique en un vecteur binaire et ceci par la mesure de ses qubits un par un.

Algorithme 4

Pour chaque Q-bit
 Générer un nombre aléatoire $r \in [0; 1]$
 Si $(r > \alpha^2)$ alors
 Nous mettons le bit à 1
 Sinon
 Nous mettons le bit à 0
 Fin
 À la fin nous obtenons un vecteur binaire.

$$\begin{bmatrix} \alpha_1 & | & \alpha_2 & | & \dots & | & \alpha_m \\ \beta_1 & | & \beta_2 & | & \dots & | & \beta_m \end{bmatrix} \xrightarrow{\text{Measure}} (0, 1, \dots, 1)$$

FIGURE 2 – Mesurer une solution.[5]

Nous prenons le vecteur binaire après la mesure pour s’assurer que toutes les solutions choisies aléatoirement sont des solutions faisables, c’est à dire la somme les poids des objets ne doit pas dépasser la capacité du sac.

Algorithme 5

```

Pour i =0 à V.size () Faire
Si = V.get(i) * Poids.get(i)
S=S + Si
Si (S < C) alors
Remplacer le bit qui contient 1 dans le vecteur V par le 0
Fin
    
```

3.6 Evaluation

Le résultat fournit par la fonction d’évaluation va permettre de sélectionner ou de refuser une solution pour ne garder que les solutions ayant le meilleur coût en fonction de la population courante : c’est le rôle de la fonction fitness. Et elle est implémentée dans une méthode CalculerFitness(), représentée par :

À condition $f(x) = \sum V_i x_i$, le $i = 1..n$ tel que $n = \text{nbre d'objets}$.

$f(x) = \sum P_i x_i \leq C$, le $i = 1..n$ tel que $n = \text{nbre d'objets}$.

Où $x = (x_1, x_2, \dots, x_n)$, x_i est 0 ou 1, p_i et v_i sont les la valeur et le poids du ième élément. C est la capacité de le sac à dos

Algorithme 6

Pour $i = 0$ à $V.size()$ Faire
 $V_i = V.get(i) * Valeurs.get(i)$
 Nous ajoutons V_i à la liste du fitness
 Trier la liste du fitness
 Garder la meilleure solution à travers la meilleure fitness dans un variable nommé b .
 Fin

3.7 MAJ de $Q(t)$ par rotation des portes quantiques pour avoir $Q(t+1)$

L'opérateur d'interférence est un opérateur d'évolution qui guide les individus vers de meilleures solutions, et finalement, vers un seul état.

En modifiant les amplitudes des individus afin d'améliorer leurs performances. Il consiste principalement à déplacer l'état de chaque qubit dans le sens de la valeur de la meilleure. Ceci est utile pour intensifier la recherche autour des meilleures solutions. Et finalement, vers un seul état.

- Il peut être effectué à l'aide d'une transformation unitaire qui permet une rotation dont l'angle est fonction des amplitudes (a_i, b_i) et de la valeur du bit correspondant dans la référence Solution.
- La valeur de l'angle de rotation $\Delta\theta$ doit être choisi de façon à éviter une convergence prématurée.
- C'est souvent déterminé empiriquement et sa direction est déterminée en fonction des valeurs de a_i, b_i et la valeur du qubit situé à la position i dans l'individu en cours de modification.

Le $[\alpha_i, \beta_i]^T$ d' i -ème Q-bit est mise à jour comme suit :

$$\begin{bmatrix} \alpha_i^{t+1} \\ \beta_i^{t+1} \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \alpha_i^t \\ \beta_i^t \end{bmatrix} \quad (3.1)$$

Tel que :

Xi	bi	f(x) > f(b)	$\Delta\theta_i$	s(ai bi)			
				ai.bi >0	ai.bi <0	ai = 0	bi = 0
0	0	0	0.001 π	-	+	\pm	\pm
0	0	1	0.001 π	-	+	\pm	\pm
0	1	0	0.08 π	-	+	\pm	\pm
0	1	1	0.001 π	-	+	\pm	\pm
1	0	0	0.08 π	+	-	\pm	\pm
1	0	1	0.001 π	+	-	\pm	\pm
1	1	0	0.001 π	+	-	\pm	\pm
1	1	1	0.001 π	+	-	\pm	\pm

FIGURE 3 – Lookup pour la rotation des portes quantiques.[23]

- ri et bi sont le ième bit de solution binaire r et le i-ème bit de la meilleure solution b respectivement.
- f est la fonction fitness.
- θ_i est l'angle de rotation, $\theta_i = s(\alpha_i, \beta_i)\Delta\theta_i$.
- s(ai bi) est le signe de l'angle de rotation .
- Nous avons utilisé une valeur dynamique de l'angle de rotation afin d'accroître la performance de l'opérateur d'interférence. Les valeurs de rotation d'angle sont indiquées dans le tableau 1.

Selon la table 1, on peut facilement remarquer que cette stratégie améliore :

- Pour chaque individu, les amplitudes de qubits mauvais selon un angle $\Delta\theta_1 = 0,08\pi$ tandis qu'il diminue, ceux qui sont bons selon un angle $\Delta\theta_2 = 0,001\pi$.
- La modification des amplitudes des qubits est fait selon les signes des amplitudes, la meilleure solution et la solution extraite par l'individu récipient.
- Il est naturel que $\Delta\theta_1 \gg \Delta\theta_2$ car les amplitudes décroissantes ne servent qu'à corriger la stochastique erreurs pour éviter une dérive génétique et pour assurer une diversité de la population.

3.8 Le Croisement

Le croisement en anglais « Cross Over », consiste à reproduire de nouvelles solutions qui hérite partiellement des solutions père. En croisant deux solutions pour avoir quatre comme résultat.

Ce croisement est similaire à un croisement classique à part qu'il opère sur des chromosomes quantiques, il génère des vecteurs de probabilités, par exemple le croisement suivant aura comme entrée les deux chromosomes S1 et S2 et produira leurs deux fils S3 et S4.

- **TauxCroisement** est un paramètre qu'on a implémenté avec fonction **Cross()** pour l'appliquer sur un nombre définie des individus et non pas toute la liste de population.

$$\begin{array}{l}
 s_1 = \left(\begin{array}{cccccc} 0.90 & 0.65 & 0.50 & 0 & 0.80 & 0.30 \\ 0.10 & 0.35 & 0.50 & 0 & 0.20 & 0.70 \end{array} \right) \\
 s_2 = \left(\begin{array}{cccccc} 0.20 & 0.10 & 0.88 & 1 & 0.70 & 0.39 \\ 0.80 & 0.90 & 0.12 & 0 & 0.30 & 0.61 \end{array} \right)
 \end{array}$$



$$\begin{array}{l}
 s_3 = \left(\begin{array}{cccccc} 0.90 & 0.65 & 0.50 & 0 & 0.80 & 0.30 \\ 0.10 & 0.35 & 0.50 & 0 & 0.20 & 0.70 \end{array} \right) \\
 s_4 = \left(\begin{array}{cccccc} 0.20 & 0.10 & 0.88 & 1 & 0.70 & 0.39 \\ 0.80 & 0.90 & 0.12 & 0 & 0.30 & 0.61 \end{array} \right)
 \end{array}$$

3.9 La Mutation

La mutation classique opère comme une petite perturbation qui inverse le bit muté,

Dans ce travail on a définie un type de mutation appelé mutation quantique qui est aussi une perturbation mais elle opère sur les probabilités d'un qubit de la solution de la façon suivante si ce qubit est représenté par $\alpha < 0 | + \beta < 1 |$, la mutation génère le qubit $\beta < 0 | + \alpha < 1 |$.

- Comme en Croisement, on a utilisé TauxMutation comme paramètre qui doit être strictement inférieur au premier taux (les taux sont des paramètres d'entrée).

Par exemple

$$s_1 = \begin{pmatrix} 0.90 & 0.65 & 0.50 & 0 & 0.80 & 0.30 & \mathbf{0.44} & 0.18 & 0.99 & 0.51 & 0 & 0.22 & 1 & 0.05 & 0.88 \\ 0.10 & 0.35 & 0.50 & 0 & 0.20 & 0.70 & \mathbf{0.56} & 0.82 & 0.01 & 0.49 & 1 & 0.78 & 0 & 0.95 & 0.12 \end{pmatrix}$$

↓

$$s_1 = \begin{pmatrix} 0.90 & 0.65 & 0.50 & 0 & 0.80 & 0.30 & \mathbf{0.56} & 0.18 & 0.99 & 0.51 & 0 & 0.22 & 1 & 0.05 & 0.88 \\ 0.10 & 0.35 & 0.50 & 0 & 0.20 & 0.70 & \mathbf{0.44} & 0.82 & 0.01 & 0.49 & 1 & 0.78 & 0 & 0.95 & 0.12 \end{pmatrix}$$

3.10 Réctification des solutions infaisables

Nous prenons les solutions binaire après la mesure pour s'assurer que toutes les solutions obtenus sont solutions faisables, cela que réaliser à condition la somme les poids des objets ne dépasse pas la capacité du sac.

Algorithme 7

```
Pour i =0 à Solutions.size () Faire
Si = Solutions.get(i) * Poids.get(i)
S=S + Si
Si (S < C) alors
Remplacer le bit qui contient 1 dans la solution par le 0.
Fin
```

3.11 Amélioration les solutions par l'algorithme de descente

Nous améliorons la meilleure solution obtenue dans chaque itération avec l'algorithme de descente.[18] Nous appliquons cet algorithme dans chaque itération sur chaque meilleure solution courante pour intensifier la recherche.

Algorithme 8

```
A = la meilleure solution.  
Pour i =0 à NombreVoisinage Faire  
A' -> Sélectionne dans voisinage ( A )  
Pour i =0 à A'.size() Faire  
On change un bit aléatoire de A  
Si ( bit aleatoire = 0) alors  
Remplacer le bit par le 1.  
Vérifier la faisabilité de la solution.  
Si ( bit aleatoire = 1) alors  
Remplacer le bit par le 0.  
Calculer fitness de A' .  
Si  $f(A') < f(A)$  Alors  
A <- A'.  
Fin
```

3.12 Conclusion

En clair, comme nous venons de le voir, ce chapitre a été consacré, à la conception de l'algorithme génétique quantique pour résoudre le problème de sac à dos qui est une succession d'opérations quantiques.

Dans le chapitre suivant, nous allons présenter nos résultats expérimentaux.

Chapitre **4**

Résultats et Expérimentation

4.1 Introduction

L'objectif de ce chapitre est de présenter les étapes de l'implémentation de l'approche proposée l'algorithme génétique quantique amélioré AGQA. Pour montrer l'efficacité notre algorithme AGQA, nous comparons ses résultats avec l'AGQ classique. Cette amélioration est effectuée en appliquant l'algorithme de la descente à chaque meilleure solution dans chaque itération.

Nous commençons tout d'abord par la présentation des ressources, l'environnement et le langage utilisés dans la réalisation de notre projet, ainsi que les données de tests, les résultats obtenus. Enfin, une comparaison entre l'approche classique des AGQ et notre approche améliorée AGQA.

4.2 Environnement de développement

4.2.1 Environnement Matériel

Afin de mener à bien ce projet, on a utilisé un ordinateur «Dell» il a les caractéristiques suivantes :

- Processeur : Intel(R) Core(TM) i3-4005U CPU @ 1.70GHz 1.70 GHz.
- Mémoire : 4GØ.
- Système d'exploitation : Windows10, 64bits.

4.2.2 Environnement logiciel

Eclipse est un environnement de développement intégré (IDE), c'est-à-dire un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aide à la programmation. Il est développé par IBM, est gratuit et disponible pour la plupart des systèmes d'exploitation, en s'appuyant principalement sur Java.

En programmant, eclipse compile automatiquement le code que vous écrivez, en soulignant en rouge ou jaune les problèmes qu'il détecte. Il souligne en rouge les parties du programme qui ne compilent pas, et en jaune les parties qui compilent mais peuvent éventuellement poser problème (on dit qu'eclipse lève un avertissement).[21]

Il est considéré comme l'un des plus célèbres environnements de développement vu les fonctionnalités et les options offertes. Pour cela, nous l'avons adopté comme outil de développement.[6]

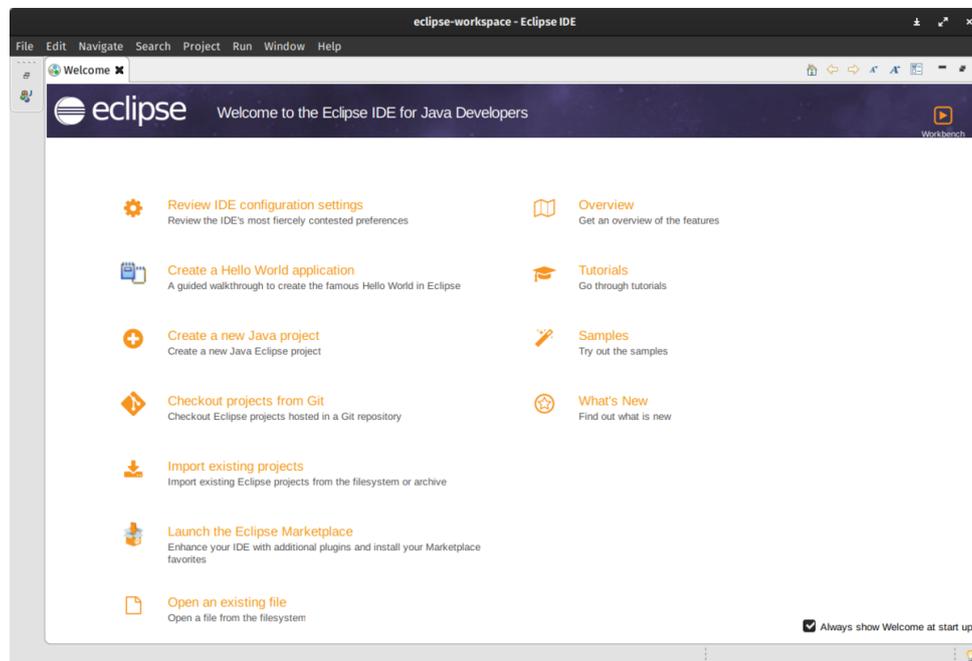


FIGURE 1 – Eclipse IDE .

4.3 Langages de programmation

Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, présenté officiellement le 23 mai 1995 au SunWorld.

Java fournit une plate-forme de développement propre et efficace basée sur les objets.[16]

La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que Unix, Windows, Mac OS ou GNU/Linux, , mais qui ont l'inconvénient d'être plus lourd à l'exécution (en mémoire et en temps processeur) à cause de sa machine virtuelle. Pour cela, divers plateformes et frameworks associés visent à guider cette portabilité des applications développées en Java.



FIGURE 2 – Java.

4.4 Présentation de l'application

Pour mieux présenter notre approche, on a créé une interface qui donne la main à l'utilisateur d'introduire les paramètres de notre algorithme.

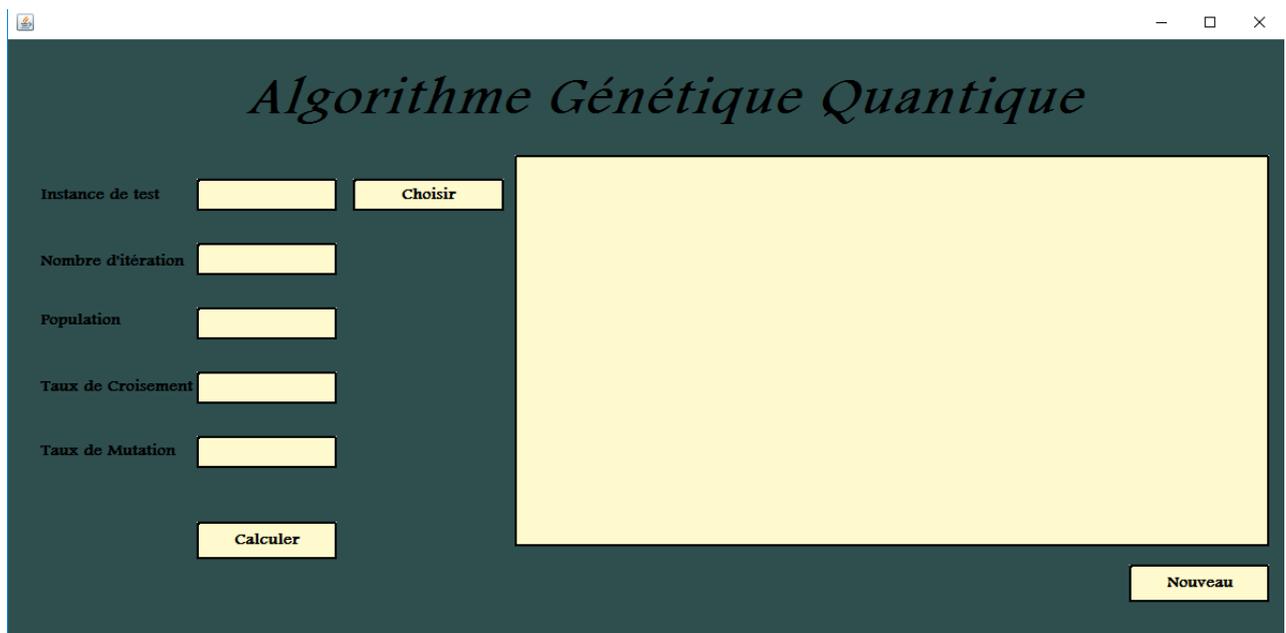


FIGURE 3 – L'interface principale.

- En cliquant sur le bouton Calculer sans ajouter les paramètres, une interface d'erreur apparaîtra.

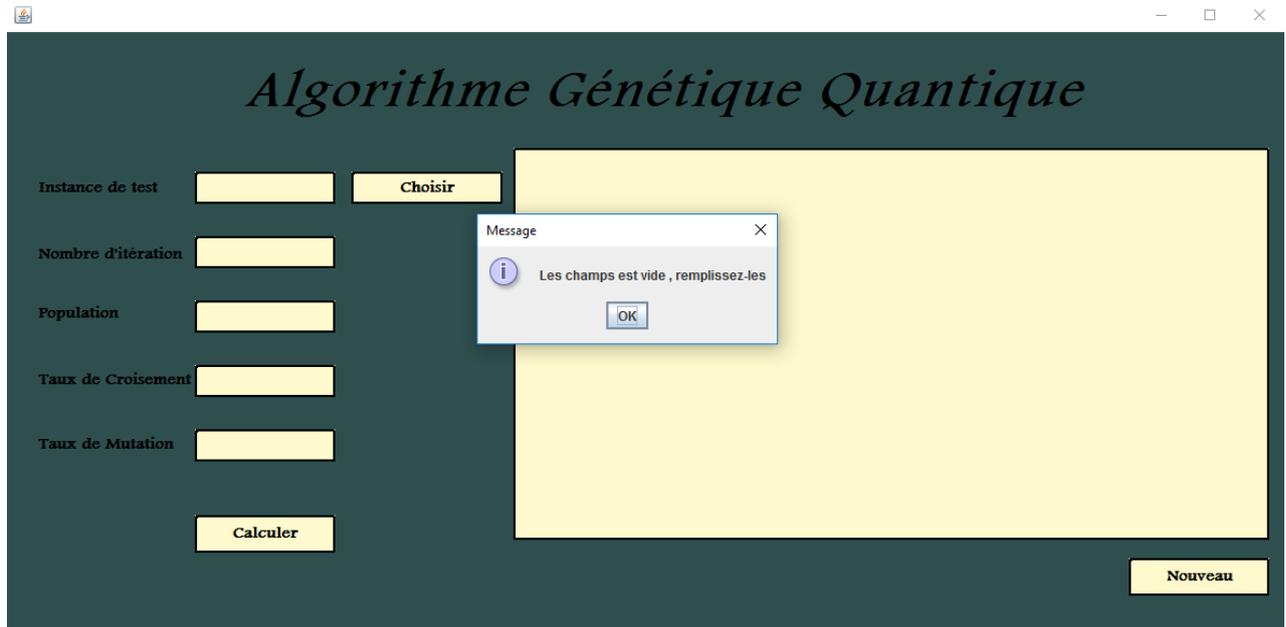


FIGURE 4 – L'interface d'erreur .

- Le bouton Calculer pour démarrer l'exécution et calculer les valeurs des objets.
- Le bouton Nouveau pour commencer un nouveau calcul.
- les champs pour remplir les paramètres de cet algorithme :
 - Instance de tests.
 - Nombre de génération.
 - Taille de la population.
 - Taux de croisement.
 - Taux du mutation.
- Pour choisir l'instance de test, on utilise le bouton choisir pour choisir un seul fichier.

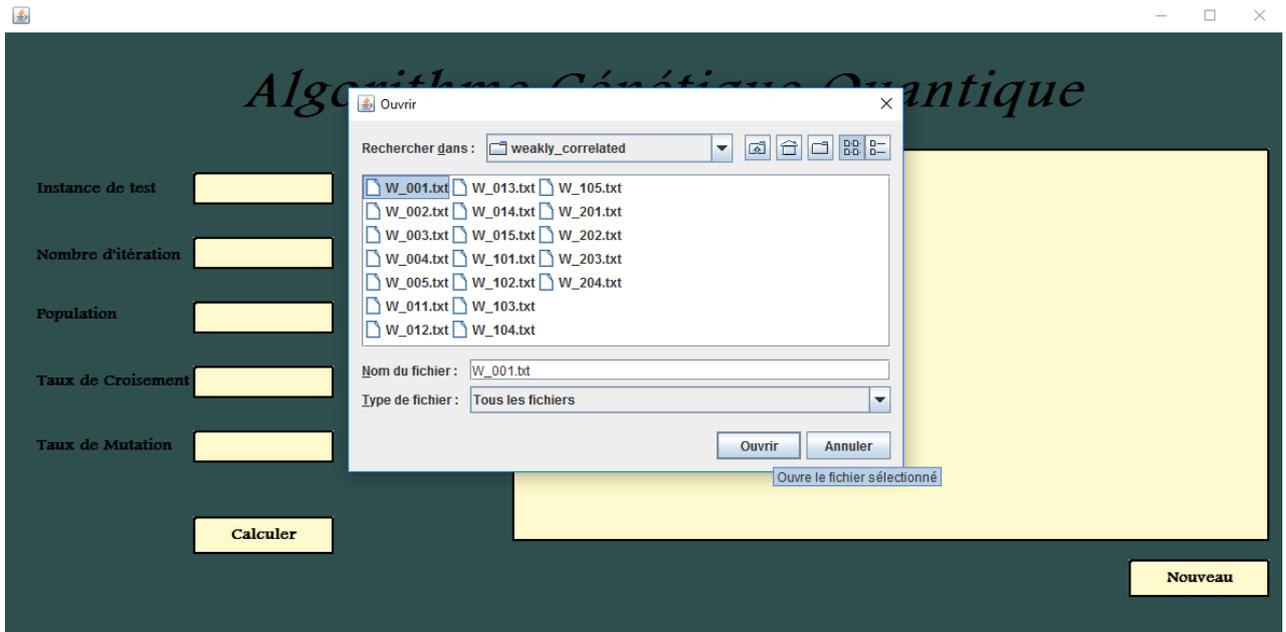


FIGURE 5 – choisir «file.txt»..

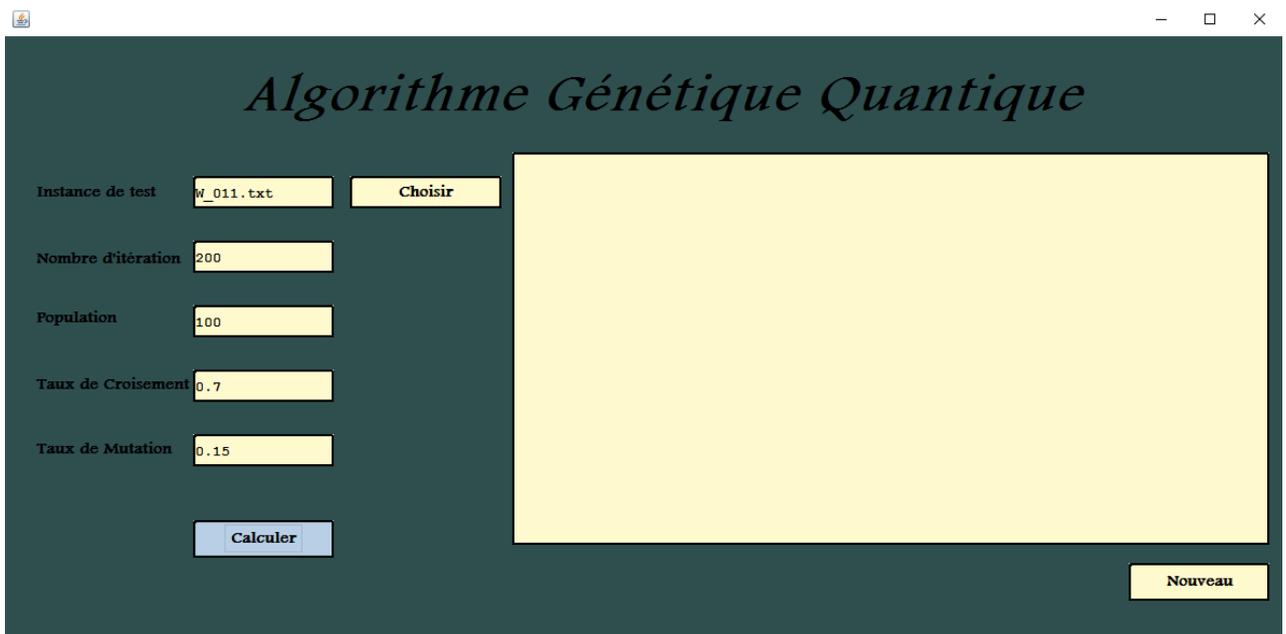


FIGURE 6 – L'interface (les champs remplis) .

- Les résultats (les valeurs des meilleures solutions, les valeurs des meilleures solutions améliorées) sont affichés dans la même interface.

4.5 Données de tests

Pour tester notre approche, nous avons utilisé les données de test téléchargeables via internet [4]. Nous utilisons les instances [4] pour le problème de sac à dos avec 50 objets, 100 objets, 200 objets et 500 objets.

pour chaque objet on a : valeur, poids. Nous gardons ces données et en plus la capacité du sac.

50 Nombre des objets	
14778 Capacité du sac	
Poids	Valeurs
845	884
758	448
421	81
259	321
512	588
485	933
784	118
384	552
477	787
584	548
909	815
505	541
282	964
756	604
619	588
251	445
910	597
983	385
811	576
903	291
311	190
730	187
899	613
684	657
473	477
181	98
435	758
611	877
914	924
967	843
478	899
866	924
261	541
806	392
549	786
15	276
720	812
399	850
825	896
669	590
2	950
494	580
868	451
244	661
326	997
871	917
192	794
568	83
239	613
968	487

FIGURE 7 – Instance Benchmark (50 objets).

4.6 Résultats expérimentaux et discussion

Dans notre application, nous testons sur des ensemble différents de deux type des instances de benchmark (Weakly_correlated, Strongly_correlated)[4], chaque type contient 20 instances, chaque instance contient entre 50 et 500 objet, notre algorithme à été exécuté plusieurs fois pour chaque instance.

Les resultats suivantes montrent les résultats expérimentaux sur le problème de sac à dos où le nombre d' objet a été fixé à 50, 100, 200 et 500. La taille de la population a été fixée à 50, 100, 200 alors que le nombre des itérations se varie de 100 jusqu'à 600.

4.6.1 Comparaison des deux méthodes approchées

Nous comparons entre les deux méthodes approchées QEA et l'approche proposée AGQA, en utilisant le GAP qui est la moyenne d'écart (pourcentage d'amélioration), ce dernier est calculé comme suit :

$$GAP_{QEA \text{ amélioré}-QEA} = \frac{FITNESS_{QEA \text{ amélioré}} - FITNESS_{QEA}}{FITNESS_{QEA \text{ amélioré}}} \times 100\%$$

les résultats sont présentés dans les tableaux 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 et 4.8. la première colonne représente le type de l'instance (Weaklly_correlated, Strongly_correlated) et le nom de l'instance (W_001, W_002, W_003, ...) ou (S_001, S_002, S_003, ...) , la deuxième colonne contient le nombre d'objet dans chaque instance, la troisième et le quatrième contiennent les résultats de l'approche QEA et l'approche QEA amélioré, la dernière colonne représente l'écart trouvé entre les solutions (GAP).

Instance		Nombre d'objet	QEA	QEA amélioré	GAP (%)
Weakly_correlated	W-001	50	16911	17100	1.10
			16911	17725	4.59
			16911	17524	3.49
			16911	17844	5.22
			16911	17228	2.38
			15273	15273	0
	Moyenne		2.79		
	W-002	50	9967	10210	2.38
			6553	7493	12.54
			12893	12893	0
			14888	15771	5.59
			14888	14888	0
			9297	10281	9.57
	Moyenne		5.01		
	W-003	50	9279	9279	0
			12272	12272	0
			12101	12101	0
			8897	10417	14.59
			12101	12318	1.76
			12318	12318	0
Moyenne		2.72			
W-004	50	135195	135195	0	
		136589	136589	0	
		139116	139499	0.27	
		136324	137426	0.86	
		100789	100789	0	
		132688	135701	2.22	
Moyenne		0.55			
W-005	50	10560	10560	0	
		13144	13144	0	
		13371	13371	0	
		13149	13166	0.12	
		14171	14307	0.95	
		12198	12198	0	
Moyenne		0.17			
Moyenne totale					2.24

TABLE 4.1 – Comparaison QEA avec QEA amélioré sur 50 objets (W_001, W_002, W_003, W_004, W_005).

Instance		Nombre d'objet	QEA	QEA amélioré	GAP (%)
Weakly_correlated	W-011	100	19129	19129	0
			21167	21167	0
			24178	24385	0.84
			24374	24456	0.33
			12539	13534	7.35
			24074	24074	0
	Moyenne		1.42		
	W-012		118061	118061	0
			232835	232835	0
			223793	228998	2.27
			204649	213798	4.27
			232933	232933	0
			232806	232806	0
	Moyenne		1.09		
	W-013		11527	11527	0
			25013	25013	0
			24357	25202	3.35
			29202	29202	0
			23947	24861	3.67
19275		20242	4.77		
Moyenne	1.96				
W-014	27898	28088	0.67		
	20188	20188	0		
	23895	23895	0		
	16998	16998	0		
	23280	23280	0		
	25835	26834	3.72		
Moyenne	0.73				
W-015	6451	7338	12		
	24753	24753	0		
	21370	22291	4.13		
	17521	18351	4.52		
	20249	20249	0		
	25300	25300	0		
Moyenne	3.44				
Moyenne totale					1.72

TABLE 4.2 – Comparaison QEA avec QEA amélioré de 100 objets (W_011, W_012, W_013, W_014, W_015).

Instance		Nombre d'objet	QEA	QEA amélioré	GAP (%)
Weakly_correlated	W-101	200	36745	37654	2.41
			38318	38600	0.73
			46357	46357	0
			37488	37488	0
			45846	45990	0.31
			43440	44437	2.24
	Moyenne		0.94		
	W-102		497624	497624	0
			500202	500202	0
			436241	445228	2.01
			498855	498855	0
			483768	486129	0.82
			503804	503804	0
	Moyenne		0.47		
	W-103		37664	38629	2.49
			33556	34522	2.79
			33212	33212	0
			42303	42303	0
			50705	50705	0
			48570	49449	1.77
Moyenne	1.17				
W-104	45520	46517	2.14		
	29753	30736	3.19		
	39043	40040	2.49		
	42105	42105	0		
	51878	51878	0		
	43098	44052	2.16		
Moyenne	1.66				
W-105	31723	32700	2.98		
	52404	53401	1.86		
	32641	32641	0		
	53344	53344	0		
	44797	44797	0		
	54070	54081	0.02		
Moyenne	0.81				
Moyenne totale					1.01

TABLE 4.3 – Comparaison QEA avec QEA amélioré de 200 objets (W_101, W_102, W_103, W_104, W_105).

Instance		Nombre d'objet	QEA	QEA amélioré	GAP (%)
Weakly_correlated	W-201	500	94655	95630	1.01
			104204	105204	0.95
			92869	93560	0.73
			128945	128945	0
			82620	82620	0
			48101	49098	1.68
	Moyenne		0.72		
	W-202		368574	378318	2.57
			976515	986328	0.99
			1288697	1288697	0
			1285084	1285084	0
			1285397	1285397	0
			1060896	1060896	0
	Moyenne		0.59		
	W-203		77631	78630	1.27
			68276	69272	1.43
			66764	66764	0
			121441	121513	0.05
			120887	120887	0
			121523	121523	0
Moyenne	0.45				
W-204	126866	126935	0.05		
	127189	127189	0		
	93911	94909	1.05		
	126909	126909	0		
	125931	125931	0		
	99621	100592	0.96		
Moyenne	0.34				
W-205	123959	123959	0		
	63746	64744	1.5		
	106069	106069	0		
	125006	125006	0		
	125552	125552	0		
	126000	126000	0		
Moyenne	0.25				
Moyenne totale					0.47

TABLE 4.4 – Comparaison QEA avec QEA amélioré de 500 objets (W_201, W_202, W_203, W_204, W_205).

Instance		Nombre d'objet	QEA	QEA amélioré	GAP (%)
Strongly_correlated	S-001	50	12032	12032	0
			10674	11657	8.43
			9909	10892	9.02
			12191	12191	0
			12198	12213	0.12
			12215	12223	0.06
	Moyenne		2.93		
	S-002	50	10255	10316	0.59
			7559	8487	10.93
			10333	10550	2.05
			10305	10453	1.41
			10457	10457	0
			5046	5964	15.39
	Moyenne		5.06		
	S-003	50	11420	11444	0.20
			11421	11501	0.69
			11418	11418	0
			11522	11522	0
			11474	11511	0.32
			11454	11502	0.41
Moyenne		0.27			
S-004	50	7552	8551	11.68	
		11967	12060	0.77	
		12074	12077	0.02	
		12035	12038	0.02	
		8760	9759	10.23	
		12032	12040	0.06	
Moyenne		3.79			
S-005	50	8739	8739	0	
		9089	9089	0	
		9208	9270	0.66	
		8764	9287	5.63	
		9129	9193	0.96	
		9230	9237	0.07	
Moyenne		1.22			
Moyenne totale					2.65

TABLE 4.5 – Comparaison QEA avec QEA amélioré de 50 objets (S_001, S_002, S_003, S_004, S_005).

Instance		Nombre d'objet	QEA	QEA amélioré	GAP (%)
Strongly_correlated	S-011	100	24988	25170	0.72
			25016	25016	0
			23568	24551	4
			19142	20092	4.72
			22151	23061	3.94
			23061	23061	0
	Moyenne		2.23		
	S-012		23290	23371	0.34
			23466	23472	0.02
			23439	23441	0.008
			23302	23367	0.27
			23384	23470	0.36
			23474	23474	0
	Moyenne		0.16		
	S-013		21339	21407	0.31
			21476	21496	0
			21353	21372	0.08
			21552	21552	0
			21457	21467	0.04
			21674	21674	0
	Moyenne		0.07		
S-014	20389	20499	0.53		
	20397	20463	0.32		
	20671	20671	0		
	21031	21089	0.27		
	18763	19761	5.05		
	20882	20964	0.39		
Moyenne	1.09				
S-015	20551	20551	0		
	7118	8111	12.24		
	17353	17353	0		
	21138	21150	0.05		
	20544	20544	0		
	20949	20949	0		
Moyenne	2.05				
Moyenne totale					1.12

TABLE 4.6 – Comparaison QEA avec QEA amélioré de 100 objets (S_011, S_012, S_013, S_014, S_015).

Instance		Nombre d'objet	QEA	QEA amélioré	GAP (%)
Strongly correlated	S-101	200	25918	26900	3.65
			11026	11026	0
			41421	41493	0.17
			41791	41791	0
			2773	3755	26.15
			41972	41998	0.06
	Moyenne		5.00		
	S-102		38119	38119	0
			33571	33571	0
			16689	16689	0
			38601	38668	0.17
			29995	30896	2.91
			38429	38429	0
	Moyenne		0.51		
	S-103		38950	39048	0.25
			38921	38921	0
			28560	29513	3.22
			39146	39146	0
			38958	38958	0
39153		39153	0		
Moyenne	0.57				
S-104	24986	24986	0		
	41322	41342	0.04		
	41360	41405	0.10		
	41789	41789	0		
	41293	41316	0.05		
	41184	41184	0		
Moyenne	0.03				
S-105	29429	30406	3.12		
	27411	28388	3.44		
	39509	39584	0.18		
	40121	40198	0.19		
	27126	27862	2.64		
	39911	39929	0.12		
Moyenne	1.61				
Moyenne totale					1.54

TABLE 4.7 – Comparaison QEA avec QEA amélioré de 200 objets (S_101, S_102, S_103, S_104, S_105).

Instance		Nombre d'objet	QEA	QEA amélioré	GAP (%)
Strongly_correlated	S-201	500	104905	105269	0.34
			105698	105698	0
			105678	105678	0
			105312	105312	0
			105757	105757	0
			106032	106032	0
	Moyenne		0.05		
	S-202		108390	108390	0
			108398	108513	0.15
			108288	108288	0
			64041	64041	0
			40821	41805	2.35
			108015	108015	0
	Moyenne		0.41		
	S-203		92269	93243	1.04
			107075	107075	0
			107356	107356	0
			107892	107892	0
			106702	106702	0
			107872	107872	0
Moyenne	0.17				
S-204	95725	96715	0		
	101481	101481	0		
	101102	101102	0		
	101504	101504	0		
	101003	101105	0.10		
	101641	101695	0.05		
Moyenne	0.025				
S-205	97075	97075	0		
	98146	98339	0.19		
	98565	98565	0		
	99895	99895	0		
	98556	98616	0.06		
	99130	99143	0.03		
Moyenne	0.04				
Moyenne totale					0.14

TABLE 4.8 – Comparaison QEA avec QEA amélioré de 500 objets (S_201, S_202, S_203, S_204, S_205).

A partir des tableaux, nous remarquons que :

Dans le premier type d'instances (Weakly_correlated) :

- Pour 50 objets que l'approche AGQA a amélioré AGQ avec une moyenne 2.24%.
- Pour 100 objets AGQA a amélioré AGQ avec un pourcentage de 1.72%.
- Pour 200 AGQA a amélioré AGQ avec un pourcentage de 1.01%.
- Pour 500 objets AGQA a amélioré AGQ avec un pourcentage de 0.47%.

Nous remarquons aussi pour la deuxième type d'instances (Strongly_correlated) :

- Pour 50 objets que l'approche AGQA a amélioré AGQ avec une moyenne 2.65%.
- Pour 100 objets AGQA a amélioré AGQ avec un pourcentage de 1.12%.
- Pour 200 objets AGQA a amélioré AGQ avec un pourcentage de 1.54%.
- Pour 500 objets AGQA a amélioré AGQ avec un pourcentage de 0.14%.

En conclusion, l'approche amélioré des AGQ a amélioré l'algorithme AGQ avec un pourcentage de 1.36%.

4.7 Conclusion

Dans ce chapitre nous avons présenté l'environnement de développement ainsi que le langage de programmation utilisé. Ensuite nous avons présenté les données de tests utilisées et présenté les résultats obtenus lors de l'application. Nous avons comparé notre 'approche proposée AGQA avec l'approche classique AGQ. Les résultats ont montré l'efficacité de cette approche pour résoudre le problème de sac à dos.

Conclusion générale

Le travail présenté dans ce mémoire a eu pour objectif d'étudier une nouvelle approche basée sur les algorithmes évolutionnaires (Algorithme génétique quantique (QGA)) et l'informatique quantique : les AEIQs. En effet, l'informatique quantique a évacué une grande partie des préoccupations des chercheurs en raison de sa puissance pour minimiser la complexité des algorithmes s'exécutant sur des machines classiques. Comme nous l'avons vu, les AEIQ sont basés sur la représentation des individus par des chromosomes quantiques. Nous avons appliqué sur ces individus les opérateurs génétiques quantiques (croisement, mutation), puis nous avons appliqué l'opérateur quantique "interférence" sur chaque individu de la population. Cette approche a été appliquée sur le fameux problème de sac à dos qui est, l'un des problèmes les plus traités dans le domaine de l'optimisation combinatoire. Nous avons amélioré notre approche en appliquant l'algorithme de descente dans chaque itération sur chaque meilleure solution.

Nos résultats expérimentaux ont montré que AEIQ (QGA) peut être une solution très prometteuse pour résoudre les problèmes d'optimisations discrets. Le problème de sac à dos, à voir, les problèmes d'optimisation discrets.

Comme perspectives, nous souhaitons améliorer la phase de réglage des paramètres en proposant une méthode approchée qui règle les paramètres en fonction de l'instance testée. Nous souhaitons ainsi de pouvoir comparer notre approche améliorée avec des méthodes récentes et valider nos résultats statistiquement.

Bibliographie

- [1] Thameur ABDELLI. “ORDINATEUR QUANTIQUE : 6. INTRODUCTION AU FONCTIONNEMENT DE L’ORDINATEUR QUANTIQUE”. In : *geneva business news* (2018).
- [2] Mohamed BATOUCHE AMER DRAA Hichem TALBI. “Une approche génétique quantique pour la fusion d’images multi-sources”. In : *Equipe vision et infographie, Laboratoire LIRE* (2004).
- [3] Karima BENATCHBA et al. “Image segmentation using quantum genetic algorithms”. In : *IECON 2006-32nd Annual Conference on IEEE Industrial Electronics*. IEEE. 2006, p. 3556-3563.
- [4] *Benchmark knapsack problem*. URL : <https://github.com/likr/kplib>.
- [5] Meryem BERGHIDA. “Stratégies adaptatives et coopératives pour la résolution de problèmes de tournées de véhicules avec collecte et livraison”. Thèse de doct. Université des Sciences et de la Technologie Houari Boumediene, 2015.
- [6] Amel BOUCHARBEB et Zakaria LABOUDI. “Implémentation des algorithmes”. In : *Université Oum El Bouaghi* (2012).
- [7] *Complexité d’un algorithme*. Laboratoire d’Informatique Gaspard Monge (LIGM) Université Paris-Est.
- [8] *Comprendre l’informatique quantique – applications métiers*. URL : <https://www.frenchweb.fr/comprendre-linformatique-quantique-applications-metiers/334236>.
- [9] *Corps noir*. URL : https://fr.wikipedia.org/wiki/Corps_noir#:~:text=En%20physique%2C%20un%20corps%20noir,dit%20rayonnement%20du%20corps%20noir..
- [10] Louis DE BROGLIE. “Recherches sur la théorie des quanta”. Thèse de doct. Migration-université en cours d’affectation, 1924.

-
- [11] Safia DJEMAME et Mohamed BATOUCHE. “Hybridation d’un Algorithme Génétique Quantique et d’un Systeme Complexe pour le Traitement d’Images”. In : *Université Ferhat Abbas, Sétif1,Algérie, Université Constantine2, Algérie* (2016).
- [12] Amira GHERBOUDJ. “Méthodes de résolution de problèmes difficiles académiques”. Thèse de doct. Université de Constantine2, 2013.
- [13] Laurent GODEFROY. “Notion de complexité algorithmique”. In : *Centre de Ressources et d’Innovation Pédagogiques de l’Université de Limoges* ().
- [14] “Google annonce une avancée majeure en informatique quantique”. In : *Radio-Canada* (2019).
- [15] *Google quantum computer leaves old-school supercomputers in the dust*. URL : <https://www.cnet.com/news/google-quantum-computer-leaves-old-school-supercomputer-in-dust/>.
- [16] James GOSLING et Henry MCGILTON. “The Java language environment”. In : *Sun Microsystems Computer Company* 2550 (1995).
- [17] Jin-Kao HAO, Philippe GALINIER et Michel HABIB. “Métaheuristiques pour l’optimisation combinatoire et l’affectation sous contraintes”. In : *Revue d’intelligence artificielle* 13.2 (1999), p. 283-324.
- [18] *Heuristiques, Méta-heuristiques*. URL : http://web4.ensiie.fr/~faye/Cnam/RCP104Heuristique_faye.pdf.
- [19] *Informatique quantique*. URL : <https://whatis.techtarget.com/fr/definition/informatique-quantique>.
- [20] *Informatique quantique : où en est-on ?* URL : <https://www.clubic.com/technologies-d-avenir/actualite-854536-informatique-quantique.html#:~:text=C'est%20ce%20qu'on,on%20parle%20alors%20d'intrication..>
- [21] *Introduction à l’utilisation d’eclipse*. URL : <https://dept-info.labri.fr/ENSEIGNEMENT/programmation2/intro-eclipse/>.
- [22] Laetitia JOURDAN. “Métaheuristiques pour l’extraction de connaissances : Application à la génomique”. Thèse de doct. Université des Sciences et Technologie de Lille-Lille I, 2003.
- [23] Zakaria LABOUDI et Salim CHIKHI. “Comparison of genetic algorithm and quantum genetic algorithm.” In : *Int. Arab J. Inf. Technol.* 9.3 (2012), p. 243-249.
- [24] Zakaria LABOUDI et Salim CHIKHI. “Evolution d’Automate Cellulaire par Algorithme Genetique Quantique.” In : *CIIA*. 2009.
- [25] *Le rayonnement du corps noir*. URL : https://fr.wikipedia.org/wiki/Rayonnement_du_corps_noir.
-

- [26] *Les Algorithmes génétiques*. 30 Janvier 2014. URL : http://igm.univ-mlv.fr/~dr/XPOSE2013/tleroux_genetic_algorithm/fonctionnement.ht.
- [27] *Les phénomènes quantiques dans l'informatique*. URL : <http://www-igm.univ-mlv.fr/~dr/XPOSE2002/Quantique/>.
- [28] Dao-Wang LI. "To Solve the Job Shop Scheduling Problem with the Improve Quantum Genetic Algorithm". In : *2012 Third Global Congress on Intelligent Systems*. IEEE. 2012, p. 88-91.
- [29] Zhengyan LIU et al. "A novel improved quantum genetic algorithm for robot coalition problem". In : *2016 IEEE International Conference on Information and Automation (ICIA)*. IEEE. 2016, p. 2061-2064.
- [30] "L'ordinateur quantique". In : *Espace de culture scientifique* (2019).
- [31] DAVID LOUAPRE. "Les 7 merveilles de la mécanique quantique". In : *Science étonnante* (2013).
- [32] *mécanique classique vs mécanique quantique*. URL : <https://forums.futura-sciences.com/physique/705749-mecanique-classique-vs-mecanique-quantique.html>.
- [33] *Mécanique quantique*. URL : https://fr.wikipedia.org/wiki/M%C3%A9canique_quantique.
- [34] *Mécanique quantique*. URL : https://fr.wikipedia.org/wiki/M%C3%A9canique_quantique#:~:text=La%20m%C3%A9canique%20quantique%20tire%20son,le%20moment%20cin%C3%A9tique%20des%20particules..
- [35] *Mécanique quantique*. 2020. URL : https://fr.wikidia.org/wiki/M%C3%A9canique_quantique.
- [36] *Optimisation Combinatoire*. juin 2013. URL : <https://www.techno-science.net/definition/6352.html>.
- [37] Aziz OUAARAB. "Résolution de Problèmes d'Optimisation Combinatoire par des Métaheuristiques Inspirées de la Nature". Thèse de doct. Université Mohammed V, Faculté des sciences-Rabat, 2015.
- [38] *Présentation des problèmes d'optimisation combinatoire*. URL : <https://www.emse.fr/~delorme/Papiers/MemoireDEA/memoire003.html#toc2>.
- [39] *Quantum*. URL : <https://research.google/teams/applied-science/quantum/>.
- [40] *Quantum Computing at IBM*. URL : <https://www.ibm.com/quantum-computing/learn/what-is-ibm-q/>.

-
- [41] *Relation de Planck-Einstein*. URL : https://fr.wikipedia.org/wiki/Relation_de_Planck-Einstein#:~:text=En%201900%2C%20le%20physicien%20allemand,1e%20ph%C3%A9nom%C3%A8ne%20du%20corps%20noir.&text=Ces%20travaux%2C%20%C3%A0%20l'origine, en%201921%20pour%20Albert%20Einstein..
- [42] Jasmeet SINGH et Mohit SINGH. “Evolution in quantum computing”. In : *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*. IEEE. 2016, p. 267-270.
- [43] Christine SOLNON. “Résolution de problèmes combinatoires et optimisation par colonies de fourmis”. In : *Université Lyon 1* (2010).
- [44] Lijuan SUN et al. “Topology control based on quantum genetic algorithm in sensor networks”. In : *Frontiers of Electrical and Electronic Engineering in China 2.3* (2007), p. 326-329.
- [45] Yangguang SUN et Mingyue DING. “Quantum genetic algorithm for mobile robot path planning”. In : *2010 Fourth International Conference on Genetic and Evolutionary Computing*. IEEE. 2010, p. 206-209.
- [46] Hichem TALBI. “Algorithmes évolutionnaires quantiques pour le recalage et la segmentation multiobjectif d’images”. Thèse de doct. Université Mentouri Constantine, 1945.
- [47] *Theorie quantique*. URL : <https://whatis.techtarget.com/fr/definition/Theorie-quantique>.
- [48] Yuefei TIAN et al. “IQGA : A route selection method based on quantum genetic algorithm-toward urban traffic management under big data environment”. In : *World Wide Web 22.5* (2019), p. 2129-2151.
- [49] G. Senizergues Y. LEROYER. “Introduction ‘a l’information quantique”. In : *ENSEIRB-MATMECA* (2017).
- [50] Yu ZHENG et al. “Quantum-inspired genetic evolutionary algorithm for course timetabling”. In : *2009 Third International Conference on Genetic and Evolutionary Computing*. IEEE. 2009, p. 750-753.
- [51] Chao ZHOU, Ping PAN et Liang HUANG. “Authenticity identification of speaker digital recording data based on quantum genetic algorithm”. In : *Multimedia Tools and Applications 77.15* (2018), p. 19399-19413.