



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Electronique
Electronique des Systèmes Embarqués

Réf. : Entrez la référence du document

Présenté et soutenu par :
Mahboub Brahim

Le : samedi 23 juin 2018

Trou noir, nouvelle heuristique d'optimisation

Jury :

Mr.	Baarir Zine –Eddine	Pr	Université Mohamed Khider de Biskra	Président
Mme.	Toumi Abida	MCA	Université Mohamed Khider de Biskra	Rapporteur
Mme.	Barkat Aicha	MAA	Université Mohamed Khider de Biskra	Examineur

الجمهورية الجزائرية الديمقراطية الشعبية

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي و البحث العلمي

Ministère de l'enseignement Supérieur et de la recherche scientifique



Université Mohamed Khider Biskra

Faculté des Sciences et de la Technologie

Département de Génie Electrique

Filière : Electronique

Option : Electronique des Systèmes Embarqués

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme:

MASTER

Thème

Trou noir, nouvelle heuristique d'optimisation

Présenté par :

Mahboub Brahim

Avis favorable de l'encadreur :

Toumi Abida

signature

Avis favorable du Président du Jury

Pr Baarir Zin-Eddine

Signature

Cachet et signature



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Electronique

Option : Electronique des Systèmes Embarqués

Thème :

Trou noir, nouvelle heuristique d'optimisation

Proposé par : Toumi Abida

Dirigé par :-Toumi Abida
-Betka Abir

Résumé

On présente dans ce mémoire une nouvelle métaheuristique; c'est un algorithme d'optimisation inspiré de phénomène trou noir cet algorithme appelé algorithme du trou noir (BHA en anglais) destiné à la résolution des problèmes d'optimisation. Il a été conçu pour trouver de bonnes solutions globales optimales sans être piégé par les optimums locaux pour les problèmes difficiles. Cette technique a été implémentée et validé par des fonctions de test standards et comparé avec deux méthodes d'optimisation, pour démontrer son efficacité. Enfin, elle a été appliquée à un problème réel ; qui est le problème de routage de véhicules. Voir pour importance et utiliser dans tous les domaines .c'est pour cela qu'on a proposé de le résoudre avec la méthode BHA qui a donné résultats acceptable.

Mots clés : les problèmes d'optimisation, les algorithmes métaheuristiques, l'algorithme du trou noir (BHA), le problème de routage de véhicule

ملخص

نقدم في هذه المذكرة خوارزمية جديدة؛ انها خوارزمية تحسين مستوحاة من ظاهرة الثقب الاسود (BHA) موجهة لحل مسائل التحسين. تم استحداثه و تصميمه لإيجاد أفضل الحلول المحسنة الشاملة من دون الوقوع أو الإنجذاب نحو الحلول المحلية في المسائل المعقدة تمت برمجة هذه التقنية و التحقق منها وذلك بتطبيقها على بعض الدوال القياسية و مقارنتها مع طريقتين للتحسين لتبيان نجاعتها وفي الأخير قمنا بتطبيقها على مشكلة حقيقية؛ الا و هي مشكلة توجيه المركبات. نظرا لاهميتها و استعمالها في العديد من المجالات ولهذا السبب اقترحنا استعمال طريقة BHA و التي أعطت نتائج مقبولة .

الكلمات المفتاحية : مسائل التحسين, الخوارزميات الاستكشافية , خوارزمية الثقب الاسود (BHA), مشكلة توجيه المركبات .

Remerciements

Tout d'abord, je remercie Allah, le tout puissant, qui m'a donné la force, la patience et la volonté pour accomplir ce modeste travail.

*Également, mon encadreur de mémoire Madame Toumi Abida et
Betka Abir*

pour son encadrement, ses directives et sa disponibilité.

*Aussi, je la remercie pour ses encouragements et sa sympathie durant toute
La période de la préparation de mon mémoire.*

*Mes remerciements s'adressent également aux membres de jury,
le professeur Baarir Zine -Eddine et madame Barkat Aicha
pour leur présence et pour le temps qu'ils ont bien voulu
consacrer pour l'évaluation de ce travail.*

*Je remercie également tous les enseignants de département d'électronique
de l'université Mohamed khider Biskra qui ont participé à ma formation
pendant tout le cycle universitaire*

*Je remercie aussi tous mes amis et mes camarades qui m'ont beaucoup
soutenu conseillé et aidé*

Dédicace

Je dédie ce modeste mémoire de Master :

Ma chère mère

Mon chère père décédé : Messaoud

Mon cher grand frère : Tofik

Ma chère sœur : Dikra

Ma chère grand-mère

À toute mes familles : MAHBOUB ET MORGHAD

À tous mes amies et collègues.

Résumé

On présente dans ce mémoire une nouvelle métaheuristique; c'est un algorithme d'optimisation inspiré de phénomène trou noir cet algorithme appelé algorithme du trou noir (BHA en anglais) destiné à la résolution des problèmes d'optimisation. Il a été conçu pour trouver de bonnes solutions globales optimales sans être piégé par les optimums locaux pour les problèmes difficiles. Cette technique a été implémentée et validé par des fonctions de test standards et comparé avec deux méthodes d'optimisation, pour démontrer son efficacité. Enfin, elle a été appliquée à un problème réel ; qui est le problème de routage de véhicules. En raison de leur importance et de leur utilisation dans tous les domaines .c'est pour cela qu'on a proposé de le résoudre avec la méthode BHA qui a donnée des résultats acceptable.

Mots clés : les problèmes d'optimisation, les algorithmes métaheuristiques, l'algorithme du trou noir (BHA), le problème de routage de véhicule

Abstract

We present in this thesis a new metaheuristic; it's an algorithm of optimization inspired by black hole phenomenon, this algorithm called algorithm of the black hole intended the resolution of the problems of optimization. It was designed to found good global optimal solutions without being trapped by the local optimum for complex problems. This technique has been implemented and validated by standard test functions and compared with two optimization methods to demonstrate its effectiveness. Finally, it has been applied to a real problem; which the problem of routing of vehicles. Given their importance and their use in all domains. That's why it was proposed to solve it with the BHA method which gave acceptable results.

Key words: the optimization problems, metaheuristic algorithms, black hole algorithm (BHA), vehicle routing problem

ملخص

نقدم في هذه المذكرة خوارزمية جديدة؛ انها خوارزمية تحسين مستوحاة من ظاهرة الثقب الاسود (BHA) موجهة لحل مسائل التحسين. تم استحداثه و تصميمه لإيجاد أفضل الحلول المحسنة الشاملة من دون الوقوع أو الإنجذاب نحو الحلول المحلية في المسائل المعقدة تمت برمجة هذه التقنية و التحقق منها وذلك بتطبيقها على بعض الدوال القياسية و مقارنتها مع طريقتين للتحسين لتبيان نجاعتها وفي الأخير قمنا بتطبيقها على مشكلة حقيقة؛ الا و هي مشكلة توجيه المركبات. نظرا لاهميتها و استعمالها في العديد من المجالات ولهذا السبب اقترحنا استعمال طريقة BHA و التي أعطت نتائج مقبولة .

الكلمات المفتاحية: مسائل التحسين, الخوارزميات الاستكشافية , خوارزمية الثقب الاسود (BHA), مشكلة توجيه المركبات .

Sommaire

Liste des Tableaux

Liste des Figures

Liste des abréviations

Introduction générale..... i

Chapitre I : La métaheuristique pour l'optimisation combinatoire

I.1. Introduction	1
I.2. Optimisation Combinatoire	1
I.3. Le problème D'optimisation Combinatoire	1
I.4. Les Méthodes d'optimisation combinatoire	3
I.4.1. Méthodes exactes (complètes)	3
I.4.2. Méthodes approchées (incomplètes)	5
I.5. Métaheuristiques	5
I.5.1. Présentation des principales Métaheuristiques	6
I.5.2. Classement des méthodes métaheuristiques	6
I.5.3. Classification des métaheuristiques	12
I.5. Exploration et exploitation des algorithmes d'optimisation	13
I.6. Conclusion	14

Chapitre II : l'optimisation inspirée du trou noir

II.1. Introduction.....	16
II.2.Phénomène du trou noir.....	17
II.3. Algorithme BHA.....	18
II.3.1. Description générale de BHA	18
II.3.2. Mécanismes de fonctionnement de méthode BHA.....	19
II.4.Avantages et Inconvénients de l'algorithme BHA.....	24
II.5. Conclusion	24

Chapitre III : Application

(optimisation de problème de routage de véhicules)

III.1 Introduction.....	26
III.2. Organigramme de l'algorithme BHA	26
III.3. les fonctions de teste.....	28
III.4 Les résultats de teste de l'algorithme BHA	30
III.4 .1 Les résultats de teste de l'algorithme BHA sur les fonctions tests	30
III.4 .2 Les résultats de teste les algorithme BHA ,SA et PSO sur les fonctions tests.....	42
III.5. Application sur le problème de routage de véhicules.....	47

III.5.1 Introduction.....	47
III.5.2 problème de routage de véhicules (VRP).....	47
III.5.3 Description du model de tournée de véhicule.....	48
III.5.4 Différents des modèles de VRP.....	48
III.5.5 Modélisation mathématique du problème.....	49
III.5.5 application de l’algorithme BH pour la résolution du problème de tourné de véhicules.....	52
III.6. Résultats de simulation.....	56
III.7. Conclusion.....	79

Conclusion générale

Bibliographie

Liste des tableaux

Tableau III.1 : Les fonctions de test.....	28
Tableau III.2 : Les courbes en 3D des fonctions de tests.....	29
Tableau III.3 : résultats de teste des fonctions par méthode BHA de l'espace de recherche [-10, 10], et la solution optimale 2D.....	30
Tableau III.4 : résultats de testé les fonctions par méthode BH de l'espace de recherche [-100, 100] et la solution optimale 2D.....	31
Tableau III.5 : résultats de testé les fonctions par méthode BH de l'espace de recherche[-1000, 1000] et .la solution optimale 2D.....	31
Tableau III.6 : résultats de testé les fonctions par méthode BHA et la solution optimale 2D.....	32
Tableau III.7: résultats de testé les fonctions par méthode BHA et la solution optimale 2D.....	32
Tableau III.8: résultats de testé les fonctions par méthode BHA et la solution optimale 2D.....	37
Tableau III.9: résultats de testé les fonctions par méthode BHA et la solution optimale 2D.....	37
Tableau III.10: résultats les valeurs de fitness par des méthodes PSO et SA et BHA sur la fonction de tests dans l'espace de recherche [-10, 10], et 100 itérations...	42
Tableau III.11: résultats des solutions optimale 2D des valeurs de fitness.....	42
Tableau III.12: résultats des valeurs de fitness par des méthodes PSO et SA et BHA sur la fonction de tests dans l'espace de recherche [-10, 10], et 500 itérations...	44
Tableau III.13: résultats des solutions optimale 2D des valeurs de fitness.....	44
Tableau III.14: résultats des valeurs de fitness par des méthodes PSO et SA et BHA sur des fonctions de tests dans l'espace de recherche [-10, 10], et 500 itérations et 5 dimension.....	46

Tableau III.15 : les paramètres de modèles 1.....	52
Tableau III.16 : les paramètres de modèles 2.....	52
Tableau III.17 : les paramètres de modèle 3.....	53
Tableau III.18 : Meilleurs coûts obtenus du 1 ^{er} modèle.....	57
Tableau III.19 : Meilleurs coûts obtenus du 2 ^{ème} modèle.....	57
Tableau III.20 : Meilleurs coûts obtenus du 3 ^{ème} modèle.....	57

Chapitre I

Figure I.1: Les minimums et maximums locaux et globaux	2
Figure I.2 : Classification des Méthodes d'optimisation combinatoire.....	3
Figure I.3: Les approches inspirées de la nature	8
Figure I.4 : Déplacement d'une particule.....	10
Figure I.5: Recherche des fourmis du plus court chemin de la nourriture vers le nid en suivant la phéromone.....	11
Figure I.6 : Classification des métaheuristiques.....	12

Chapitre II

Figure II.1 : Vue schématique d'un trou noir	17
Figure II.2 : Le trou noir et la distribution les stars dans l'espace	17
Figure II.3. : Vue les éléments d'horizon des événements.....	22
Figure. II.4. : Algorithme du trou noir (BHA).....	23

Chapitre III

Figure III.1: Les étapes de l'algorithme du trou noir.....	27
Figure III.2: variation de la solution optimale en fonction d'itérations (2D)	33
Figure III.3: variation de la solution optimale en fonction d'itérations (5D)	33
Figure III.4: variation de fitness en fonction d'itérations (2D)	33
Figure III.5: variation de fitness en fonction d'itérations (5D)	33
Figure III.6: variation de rayon en fonction d'itérations (2D)	34

Figure III.7: variation de rayon en fonction d'itérations (5D).....	34
Figure III.8: variation de la solution optimale en fonction d'itérations (2D)	34
Figure III.9: variation de la solution optimale en fonction d'itérations (5D)	34
Figure III.10: variation de fitness en fonction d'itérations (2D)	35
Figure III.11: variation de fitness en fonction d'itérations (5D)	35
Figure III.13: variation de rayon en fonction d'itérations (5D).....	36
Figure III.14: variation de la solution optimale en fonction d'itérations (20 itérations).	38
Figure III.15: variation de la solution optimale en fonction d'itérations (20 itérations).....	38
Figure III.16: variation de fitness en fonction d'itérations (20 itérations)	39
Figure III.17: variation de fitness en fonction d'itérations (50 itérations).....	39
Figure III.18: variation de rayon en fonction d'itérations (20 itérations)	39
Figure III.19: variation de rayon en fonction d'itérations (50 itérations).....	39
Figure III.20: variation de la solution optimale en fonction d'itérations (20 itérations).	40
Figure III.21: variation de la solution optimale en fonction d'itération (20 itérations).....	40
Figure III.22: variation de fitness en fonction d'itérations (20 itérations)	41
Figure III.23: variation de fitness en fonction d'itérations (50 itérations).....	41
Figure III.24: variation de rayon en fonction d'itérations (20 itérations)	41

Figure III.25: variation de rayon en fonction d'itérations (50 itérations).....	41
Figure III.26: Résultats de variation de fitness en fonction d'itérations (100 itérations) sur la fonction Alpine par les trois méthodes	43
Figure III.27: la figure ci-dessus, pour éclairé les solutions optimales.....	44
Figure III.28: Résultats de variation de fitness en fonction d'itérations (500 itérations) sur la fonction Alpine par les trois méthodes.....	45
Figure III.29: la figure ci-dessus, pour éclairé les solutions optimales.....	46
Figure III.30: La différence entre le problème TSP et VRP	47
Figure III.31 : Présentation graphique du problème de tournées de véhicules VRP.....	48
Figure III.32 : Organigramme d'optimisation par BHA.....	56
Figure III.33: Résultats de variation de fitness en fonction d'itérations (50 itérations) et la forme de modèle 1.....	58
Figure III.34: courbes comparatives des coûts en fonction d'itérations pour les différents méthodes (PSO, SA, BHA).....	59
Figure III.35: Résultats de variation de fitness en fonction d'itérations (200itérations et la forme de modèle 1.....	60
Figure III.36: courbes comparatives des coûts en fonction d'itérations pour Les différents méthodes (PSO, SA, BHA).....	61
Figure III.37: Résultats de variation de fitness en fonction d'itérations (1000 itérations) et la forme de modèle 1.....	61
Figure III.38: courbes comparatives des coûts en fonction d'itérations pour les différents méthodes (PSO, SA, BHA).....	61
Figure III.39: Résultats de variation de fitness en fonction d'itérations (3000itérations) et la forme de modèle 1.....	62
Figure III.40: courbes comparatives des coûts en fonction d'itérations pour les différents méthodes (PSO, SA, BHA).....	62

Figure III.41: Résultats de variation de fitness en fonction d'itérations (50 itérations) et la forme de modèle 2.....	64
Figure III.42: courbes comparatives des coûts en fonction d'itérations pour les différents méthodes (PSO, SA, BHA).....	65
Figure III.43: Résultats de variation de fitness en fonction d'itérations (200itérations) et la forme de modèle 2.	66
Figure III.44: Résultats courbes comparatives des coûts en fonction d'itérations pour les différents méthodes (PSO, SA, BHA).....	67
Figure III.45: Résultats de variation de fitness en fonction d'itérations (1000 itérations) et la forme de modèle 2.....	67
Figure III.46: courbes comparatives des coûts en fonction d'itérations pour les différents méthodes (PSO, SA, BHA).....	68
Figure III.47: Résultats de variation de fitness en fonction d'itérations (3000itérations) et la forme de modèle 2.	69
Figure III.48: courbes comparatives des coûts en fonction d'itérations pour les différents méthodes (PSO, SA, BHA).....	70
Figure III.49: Résultats de variation de fitness en fonction d'itérations (50 itérations) et la forme de modèle 3.....	71
Figure III.50: courbes comparatives des coûts en fonction d'itérations pour les différents méthodes (PSO, SA, BHA).....	72
Figure III.51: Résultats de variation de fitness en fonction d'itérations (200itérations) et la forme de modèle 3.....	73
Figure III.52: courbes comparatives des coûts en fonction d'itérations pour les différents méthodes (PSO, SA, BHA).....	74
Figure III.53 Résultats de variation de fitness en fonction d'itérations (1000 itérations et la forme de modèle 3.....	75
Figure III.54: courbes comparatives des coûts en fonction d'itérations pour les différents méthodes (PSO, SA, BHA).....	76
Figure III.55: Résultats de variation de fitness en fonction d'itérations (3000itérations) et la forme de modèle 3.	77
Figure III.56: courbes comparatives des coûts en fonction d'itérations pour les différents méthodes (PSO, SA, BHA).....	78

Liste des abréviations

PSO : **P**articule **S**warm **O**ptimization

ACO : **A**nt **C**olony **O**ptimization

BHA : **B**lack **H**ole **A**lgorithme

SA : **S**imulated **A**nnealing

ST : **T**abu **S**earch

TSP : **T**ravelling **S**alesman **P**roblem

VRP : **V**ehicle **R**outing **P**roblem

CVRP : **C**apacitated **V**ehicle **R**outing **P**roblem

MDVR : **M**ultiple **D**epot **V**RP

2D : 2 **d**imensions

5D : 5 **d**imensions

Introduction générale

Les applications de l'optimisation sont nombreuses, où chaque processus peut-être amélioré et optimisé. Donc, il n'y a aucune entreprise ou société qui n'est pas impliquée dans la résolution de problèmes d'optimisation. En effet, de nombreuses applications difficiles ou complexes peuvent être modélisées comme des problèmes d'optimisation. L'objectif de l'optimisation est de minimiser le temps de calcul, le coût, la distance ...etc.

Ces problèmes ne peuvent pas être résolus par des méthodes exactes dans un laps de temps raisonnable, ainsi, ils sont utilisées pour trouver au moins une solution optimale d'un problème c'est-à-dire qu'il n'existe pas d'autres solutions pour lesquelles la fonction objective a un meilleur score. Les méthodes exactes les plus réussies appartiennent aux paradigmes de la programmation dynamique, de la programmation linéaire en nombres entiers, ou des méthodes de recherche arborescente (Branch & Bound). Alors, les méthodes approchées sont la principale alternative pour résoudre cette catégorie de problèmes. Ces algorithmes d'approximation sont divisés en deux classes : méthodes heuristiques et méthodes métaheuristiques.

Méthodes heuristiques sont des méthodes de rechercher d'une solution optimale pour un problème particulier. Elles apportent une solution à un problème d'optimisation combinatoire en un temps raisonnable sans garantir l'optimalité. L'heuristique offre des avantages par rapport à une méthode exacte qui est souvent beaucoup plus lente qu'une méthode heuristique.

Méthodes métaheuristiques sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global. Elles sont inspirées par des phénomènes naturels, qu'ils soient en physique ou en biologie de l'évolution ou encore en éthologie (recuit simulé, algorithme génétique, essais particuliers ...etc), elles peuvent s'appliquer à différents problèmes. Il existe un grand nombre de métaheuristiques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Elles peuvent être basées sur une solution unique ou sur une population de solutions. Les méthodes à une solution unique : partent d'une solution initiale, intensifient la recherche en exploitant une partie de l'espace de recherche en effectuant une descente vers une solution optimale.

Les méthodes à une population de solution : assumer des solutions aléatoire, ensuite le développer progressivement dans l'espace de recherche pour atteindre la meilleure solution, elles sont utilisés pour optimiser les problèmes complexes de grandes tailles.

Dans ce mémoire, nous présentons algorithme du trou noir (BHA) on va l'implémenter, le tester, et puis on va l'appliquer à un problème de routage de véhicules.

Notre mémoire est organisé comme suit :

le premier chapitre : donne une présentation des notions de base sur l'optimisation combinatoires et une classification générale des méthodes d'optimisation. On mettant l'accent sur les classes .

Le deuxième chapitre : le principe de l'algorithme du trou noir (BHA) est détaillé dans ce chapitre, en mentionnant aussi les processus et les équations mathématiques de base de cet algorithme.

Le troisième chapitre : nous allons appliquer la technique BHA sur quelques fonctions tests connues, et présenté des résultats de test .Ensuite défini le problème de routage de véhicules et leurs types. et modélisé mathématique le problème à résoudre .Puis nous envisageons de l'appliquer à un problème réel, qui consiste le problème de routage de véhicules avec contrainte de capacité de véhicule (CVRP), et présenté des résultats de notre application, et comparer avec les deux techniques d'optimisation SA et PSO.

A la fin, nous présenterons une conclusion générale et quelques perspectives envisageables.

Chapitre I

La métaheuristique pour l'optimisation combinatoire

Chapitre I

La métaheuristique pour l'optimisation combinatoire

I.1. Introduction

L'optimisation combinatoire occupe une place très importante en plusieurs domaines comme recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation.

Ce chapitre présente les concepts généraux de l'optimisation combinatoire, et les problèmes d'optimisation et les méthodes de résolution . Par ailleurs, nous nous intéressons à la classification en général des méthodes d'optimisation combinatoire en citant des méthodes existantes.

I.2. Optimisation Combinatoire

L'optimisation combinatoire est une technique mathématique, qui consiste à minimiser ou maximiser une fonction objectif (Coût, temps, distance, etc.). Dont le but est de rechercher et définir une solution optimale la plus appropriée pour l'optimisation à partir d'un ensemble de solutions possible. [1]

I.3. Le problème D'optimisation Combinatoire

Un problème d'optimisation est défini par un ensemble de variables, une fonction objective sous certaines contraintes. L'espace de recherche est l'ensemble des solutions possibles du problème. Résoudre un problème d'optimisation consiste à trouver la meilleure solution soit en minimisant ou maximisant la fonction objective de problème posé. [2]

Soit n un entier strictement positif et soient :

$$D \subset R^n \text{ un sous-ensemble non vide de } R^n, \text{ et}$$
$$f : D \rightarrow R \text{ une application sur } D \text{ à valeurs réelles.}$$

Un problème d'optimisation consiste à déterminer, lorsqu'il existe, un extremum, minimum ou maximum, de f sur D . On note un tel problème :

$$\min_{x \in D} f(x) \quad \text{ou} \quad \max_{x \in D} f(x)$$

- un minimum (ou minimum global) x^* de f sur D est un point $x^* \in D$, tel que $\forall x^* \in D$,

$$f(x^*) < f(x)$$

- un maximum (ou maximum global) x^* de f sur D est un point $x^* \in D$, tel que $\forall x^* \in D$,

$$f(x^*) > f(x) \quad [3]$$

Pour un problème de minimisation ou maximisation on cherche à déterminer une solution x qui minimise ou maximise la fonction objective. Mais peut effectuer une convergence prématurée, c-à-d il converge vers un optimum local avant d'aboutir l'optimum global.

Nous pouvons distinguer deux types d'optimum locaux

Minimum Local : une solution x est minimum local par rapport à une structure de voisinage N si $\forall x^* \in N(x) f(x^*) > f(x)$.

Maximum Local : une solution x est maximum local par rapport à une structure de voisinage N si $\forall x^* \in N(x) f(x^*) < f(x)$

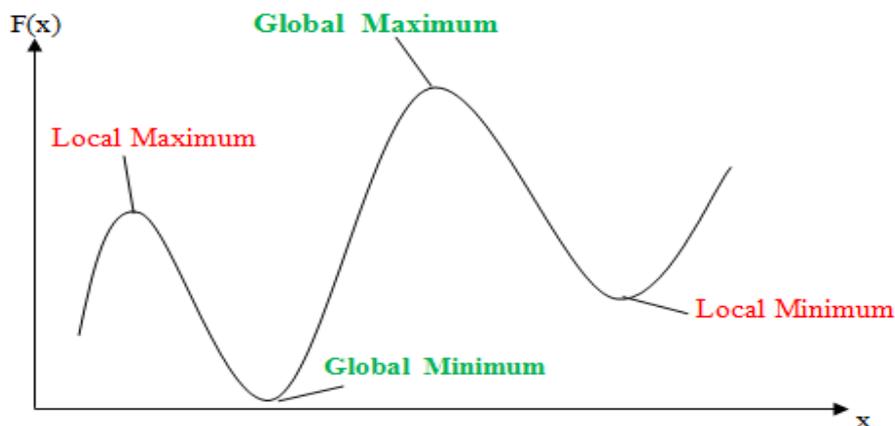


Figure I.1: Les minimums et maximums locaux et globaux

I.4. Les Méthodes d'optimisation combinatoire

Pour définir une méthode de résolution on doit d'abord étudier la complexité du problème, en général les problèmes peuvent être résolus par un algorithme exact ou un algorithme approximatif. Avec les algorithmes exacts, les solutions sont optimales et leur optimalité est garantie. Les algorithmes approximatifs (ou heuristiques) génèrent des solutions de grande qualité dans un temps raisonnable, mais il n'y a aucune garantie de trouver une solution optimale globale. [4]

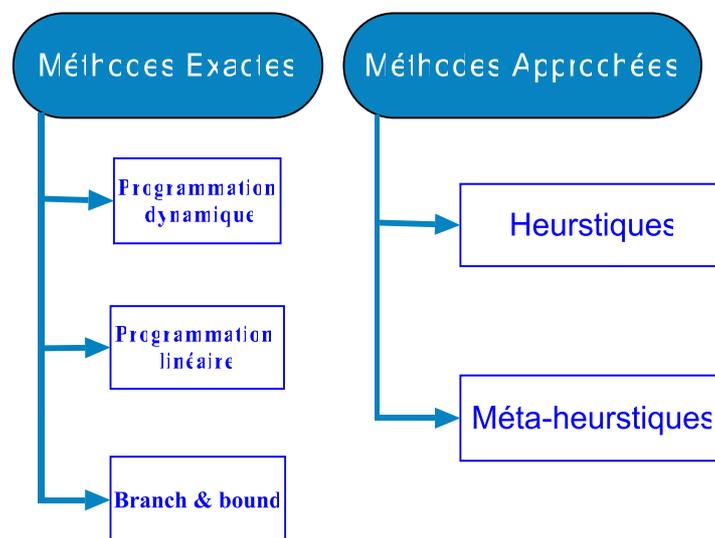


Figure I.2 : Classification des Méthodes d'optimisation combinatoire

I.4.1. Méthodes exactes (complètes) :

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 de leur réponse en un temps fini. Grâce à ces méthodes on peut trouver des solutions optimales. Mais ces méthodes s'avèrent, malgré les progrès réalisés, plutôt inefficaces à mesurer, que la taille du problème devient importante. Dans cette classe des méthodes exactes, on peut trouver les algorithmes classiques suivants : La programmation dynamique, La programmation linéaire, Les méthodes de recherche arborescente (Branch & bound). [4]

❖ La programmation dynamique :

La résolution d'un problème en programmation dynamique est basée sur une décomposition du problème en sous-problèmes plus simples. A chaque sous-problème correspond un ensemble d'options, représentant chacune un coût en terme de fonction objectif. Les algorithmes basés sur la programmation dynamique sont généralement faciles à implémenter et très efficaces pour résoudre les problèmes de petites et moyennes tailles.[16]

❖ La programmation linéaire :

C'est une branche de l'optimisation permettant de résoudre de nombreux problèmes économiques et industriels. Se posent lorsque l'on cherche à rendre optimale (minimum ou maximum) une fonction linéaire de plusieurs variables, ces variables étant assujetties à des contraintes linéaires, c'est à dire, du premier degré.

❖ Les méthodes de recherche arborescente (Branch & bound) :

C'est une des méthodes qui propose un mécanisme de recherche très intelligent, grâce à lequel elle permet une bonne exploitation de l'espace de recherche et l'aboutissement à la solution optimale plus rapidement que d'autres méthodes exactes en combinant deux principes primordiaux: la séparation et l'évaluation.[9]

Les méthodes de résolution exactes possèdent quelques avantages et inconvénients:

a) Les avantages

- Elles sont utilisées pour trouver au moins une solution optimale d'un problème donné.
- Elles procèdent par énumération de toutes les solutions possibles afin de trouver la meilleure solution.

b) Les inconvénients

- Utilisation pratique difficile (explosion combinatoire).
- Résolution des problèmes bi-critères de petites tailles. [5]

I.4.2. Méthodes approchées (incomplètes) :

Une méthode approchée est une méthode d'optimisation qui a pour but de trouver une solution réalisable de la fonction objective en un temps raisonnable, mais sans garantie d'optimalité. L'avantage principal de ces méthodes est qu'elles peuvent s'appliquer à n'importe quelle classe de problèmes, faciles ou très difficiles. [6]

Ils peuvent être classés en deux catégories : les heuristiques et les métaheuristiques. Les heuristiques sont des méthodes approchées et rapides pour résoudre un problème donné. Mais cette méthode est incapable d'optimiser autre problème, c'est-à-dire spécifique seulement pour résoudre un problème donné.[4]

La métaheuristique est une nouvelle génération de méthodes approchées puissantes et générales, qui peuvent être appliquées pour résoudre presque n'importe quel problème d'optimisation.

I.5. Métaheuristiques

Avant d'expliquer et définir le mot « méta-heuristique » comme méthode de la résolution des problèmes d'optimisation on observe que ce dernier mot se décompose en deux mots grecs : heuristique qui vient de verbe heurisken et qui signifie « trouver » et Meta qui est un préfixe signifiant « au-delà » ou « plus haut ». Les méta-heuristiques sont apparues dans les années 1980 et forment une famille d'algorithmes d'optimisation visant à résoudre des problèmes d'optimisation difficile. Elles sont généralement utilisées comme des méthodes génériques pouvant optimiser une large gamme de problèmes différents, sans nécessiter de changements profonds dans l'algorithme employé. [7]

La métaheuristique c'est une stratégie de niveau supérieur visant à faire face à différents types des problèmes et de guider la recherche d'une solution optimale. C'est leurs caractéristiques aléatoire et itératif, qui en font une méthode plus efficace afin de déterminer des solutions optimales.

I.5. 1. Présentation des principales Métaheuristiques

Il existe un grand nombre des métaheuristiques désignées pour l'optimisation. Nous allons nous appuyer sur la classification qui distingue les méthodes de solution unique (qui exploitent séquentiellement un seul voisinage) et des méthodes basées sur des populations de solutions (qui exploitent plusieurs à la fois). [8]

I.5.2. Classement des méthodes métaheuristiques

a) Les métaheuristiques à solution unique

Les métaheuristiques à base de solution unique, aussi appelées méthodes de trajectoire. Contrairement aux métaheuristiques à base de population [8], ces méthodes débutent la recherche avec une seule solution initiale. Elles se basent sur la notion du voisinage pour améliorer la qualité de la solution courante. En fait, la solution initiale subit une série de modifications en fonction de son voisinage. Nous présenterons ici les méthodes les plus utilisées et leur utilisation en extraction de connaissances : les méthodes de descente, le recuit simulé et la recherche tabou. [9]

❖ Les méthodes de descente (Hill Climbing)

La méthode de la descente (la méthode de recherche locale) passe d'une solution à une autre dans l'espace des solutions candidates (l'espace de recherche) qu'on note S , jusqu'à ce qu'une solution considérée comme optimale soit trouvée ou que le temps imparti soit dépassé. La méthode de recherche locale la plus élémentaire est la méthode de descente. L'inconvénient majeur de la méthode de descente est son arrêt au premier minimum local rencontré. [10]

❖ La recherche tabou

La recherche tabou (TS) est une méthode de recherche locale combinée avec un ensemble de techniques permettant d'éviter d'être piégé dans un minimum local. Cette méthode a montré une grande efficacité pour la résolution des problèmes d'optimisation difficiles. En effet, à partir d'une solution initiale s dans un ensemble de solutions local S , des sous-ensembles de solution $N(s)$ appartenant au voisinage S sont générés. Par

l'intermédiaire de la fonction d'évaluation nous retenons la solution qui améliore la valeur de f , choisie parmi l'ensemble de solutions voisines $N(s)$. [10]

❖ **Le recuit simulé (Simulated Annealing)**

Le recuit simulé en anglais « Simulated Annealing » est souvent présenté comme la plus ancienne des métaheuristicues. Il s'inspire d'un processus utilisé en métallurgie.

La force de cet algorithme réside dans sa capacité à éviter de chute dans des extrêmes locaux. Pour acceptant des solutions de moindre qualité dans l'espoir de sortir d'un optimum local. Cette acceptation se fait suivant la probabilité donnée par $e^{(-\frac{\delta}{T})}$, où δ est la différence entre la valeur de l'objectif de la nouvelle solution et la valeur de l'objectif de la solution courante et T est la température est un paramètre fixé au début de l'algorithme et diminue graduellement tout au long de l'exécution de l'algorithme. Plus la valeur T est élevée, plus une solution de qualité moindre a de chances d'être acceptée. Plus T s'approche de 0, moins une solution de mauvaise qualité a de chances d'être acceptée. À chaque degré de température, un nombre prédéfini de solutions voisines est visité. Une fois cette quantité atteinte, la température est diminuée. La meilleure façon de diminuer la température. [10] [15]

b) Les métaheuristicues à population de solutions

Les métaheuristicues (les algorithmes évolutionnaires, l'optimisation par colonie de fourmis, l'optimisation par essaim de particules, ... etc.) sont utilisés pour optimiser les problèmes complexes de grandes tailles. Contrairement aux méthodes exactes, les métaheuristicues à population de solutions améliorent une population de solutions, au fur et à mesure des itérations. [11]



Figure I.3: Les approches inspirées de la nature .

Ces méthodes offrent des solutions sous optimales acceptables dans des temps raisonnables, car il n'y a aucune garantie théorique que ces approches arrivent à la solution globale. Par ailleurs, les métaheuristiques sont très efficaces dans la résolution de nombreux problèmes à savoir la robotique, l'aérodynamique, l'automobile, l'électronique et la bioinformatique, ... etc.

❖ L'optimisation par l'algorithme génétique

L'algorithme génétique représente une célèbre métaheuristique évolutionnaire. Il a été proposé par Jhon Holland en 1975 [Holland, 1975]. L'algorithme génétique s'inspire des mécanismes biologiques tels que les lois de Mendel et la théorie de l'évolution proposée par Charles Darwin [Darwin, 1859]. Son processus de recherche de solutions à un problème donné imite celui des êtres vivants dans leur évolution. Il utilise le même vocabulaire que celui de la biologie et la génétique classique, on parle donc de: gène, chromosome, individu, population et génération. [9]

L'algorithme génétique est basé sur trois éléments principales

La Sélection : Cette opération s'intervient dans deux phases. Elle est appliquée au premier lieu pour choisir les meilleurs individus parents qui vont se reproduire pour construire de nouveaux individus enfants. Ensuite, elle est appliquée à la fin de chaque itération pour opter pour les individus qui vont survivre et construire la nouvelle population.

La reproduction : cette opération est en général composée de deux autres opérations : le croisement et la mutation. Elle permet la génération de nouveaux individus en combinant (phase de croisement), les caractéristiques des individus sélectionnés puis en appliquant quelques modifications de certains individus (phase de mutation) pour améliorer leurs qualités.

L'évaluation : Cette opération consiste à mesurer la qualité de chaque individu (calculer la fitness des individus). [9]

❖ L'optimisation par essais particulaire PSO

L'Optimisation par Essaim Particulaire (PSO en anglais) a été proposée par (Kennedy et Eberhart 1995). Cette méthode est inspirée du comportement social des animaux évoluant en essaim. Au départ, ils cherchaient à simuler la capacité des oiseaux à voler de façon synchrone et leur aptitude à changer brusquement de direction tout en restant en une formation optimale.

Les particules sont les individus et elles se déplacent dans l'hyperespace de recherche en se basant sur des informations limitées :

1. Chaque particule est dotée d'une mémoire qui lui permet de mémoriser le meilleur point par lequel elle est déjà passée et elle a la tendance à retourner vers ce point.
2. Chaque particule est informée du meilleur point connu au sein de son voisinage et elle va tendre à aller vers ce point. [12]

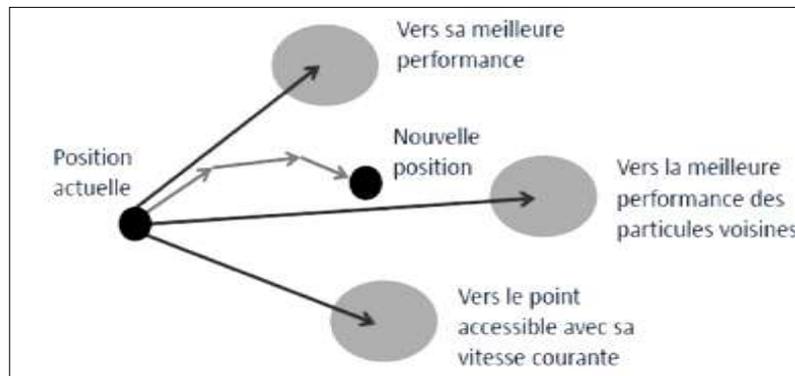


Figure I.4 : Déplacement d'une particule.

❖ L'optimisation par colonie de fourmis (ACO)

Les algorithmes de colonies de fourmis ont été proposés par Colorni, Dorigo et Maniezzo en 1992 (Colorni et al. 1992b). L'idée de base imite le comportement collectif des fourmis lors de leur déplacement entre la fourmilière et la source de nourriture. Dans l'objectif de rechercher la nourriture en parcourant le plus court chemin, Quand une fourmi atteint la source de nourriture, elle retrace son chemin de retour vers le nid par une quantité de phéromone pour avertir les autres fourmis qu'il y a de la nourriture à la fin du chemin. Les fourmis les plus rapidement arrivées au nid après avoir visité la source de nourriture sont celles qui empruntent le chemin le plus court, aussi la quantité de phéromone présente sur le plus court chemin est plus importante que celle présente sur le chemin le plus long. Une piste qui présente une plus grande concentration en phéromone est plus attirante par les fourmis, elle a une probabilité plus grande d'être empruntée. La piste courte va alors être plus renforcée que la plus longue, et elle sera choisie par la quasi-totalité des fourmis. L'idée est de représenter le problème à résoudre sous la forme de la recherche d'un meilleur chemin dans un graphe, puis d'utiliser des fourmis artificielles pour rechercher de bons chemins dans ce graphe.[8]

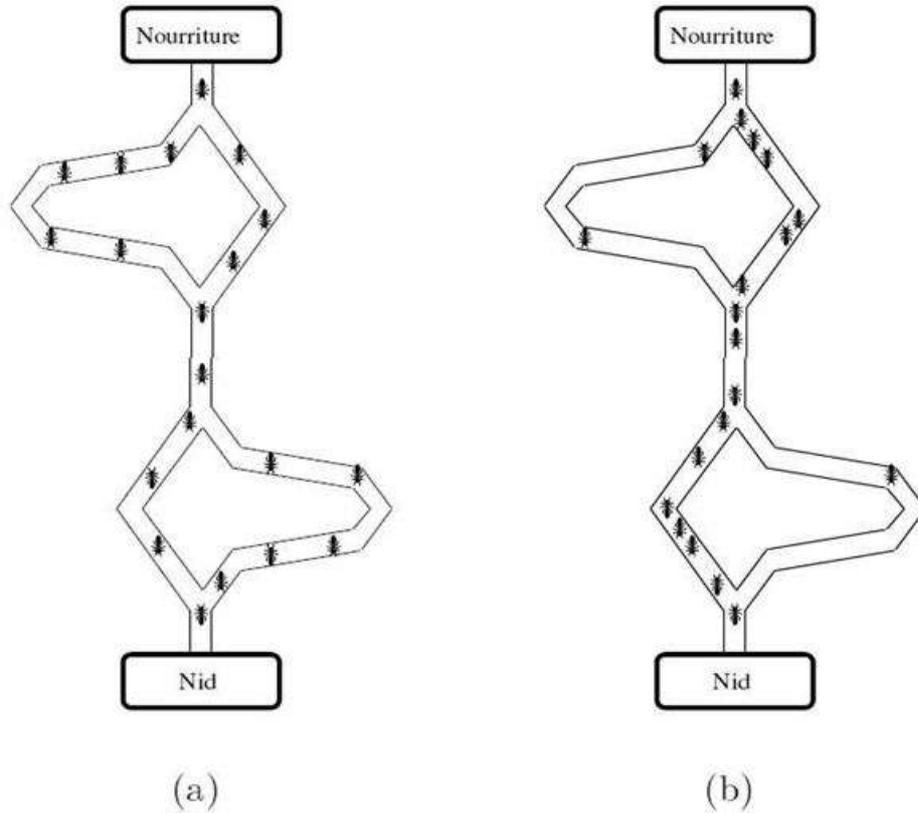


Figure I.5: Recherche des fourmis du plus court chemin de la nourriture vers le nid en suivant la phéromone.[12]

❖ **L'optimisation par l'algorithme du trou noir (BHA)**

L'algorithme du trou noir a été proposé par Abdolreza Hatamlou. Cette méthode est inspirée des mouvements du trou noir dans l'espace et les mouvements des stars autour de lui où le trou noir est caractérisé par le rayon d'horizon. Le phénomène de trou noir se manifeste par le fait que quelques soit les stars à l'intérieur de ce rayon vont être absorbé et disparaissent.

I.5.3. Classification des métaheuristiques

Plusieurs classes de métaheuristiques peuvent être identifiées :

- Inspiration de la nature ou non.
- Utilisation de l'historique de la recherche (mémoire).
- La manipulation d'une solution à la fois ou un ensemble des solutions dites population

On classe les métaheuristiques selon leur principe de fonctionnement :

- Les méthodes constructives : construisent une solution progressivement en prenant à chaque étape une décision (par exemple, prochaine ville à visiter dans la tournée d'un voyageur de commerce) qui ne sera jamais remise en cause.
- Méthodes de recherche locale : méthodes d'amélioration itératives qui se déplacent pas à pas dans le « voisinage » de la solution courante en cherchant des « pistes prometteuses » vers la solution optimale.
- Méthodes évolutives : manipulent un groupe de solutions admissibles à chaque étape de recherche plutôt qu'une seule solution, s'inspirent de phénomènes naturels (algorithmes génétiques, fourmis...). [13]

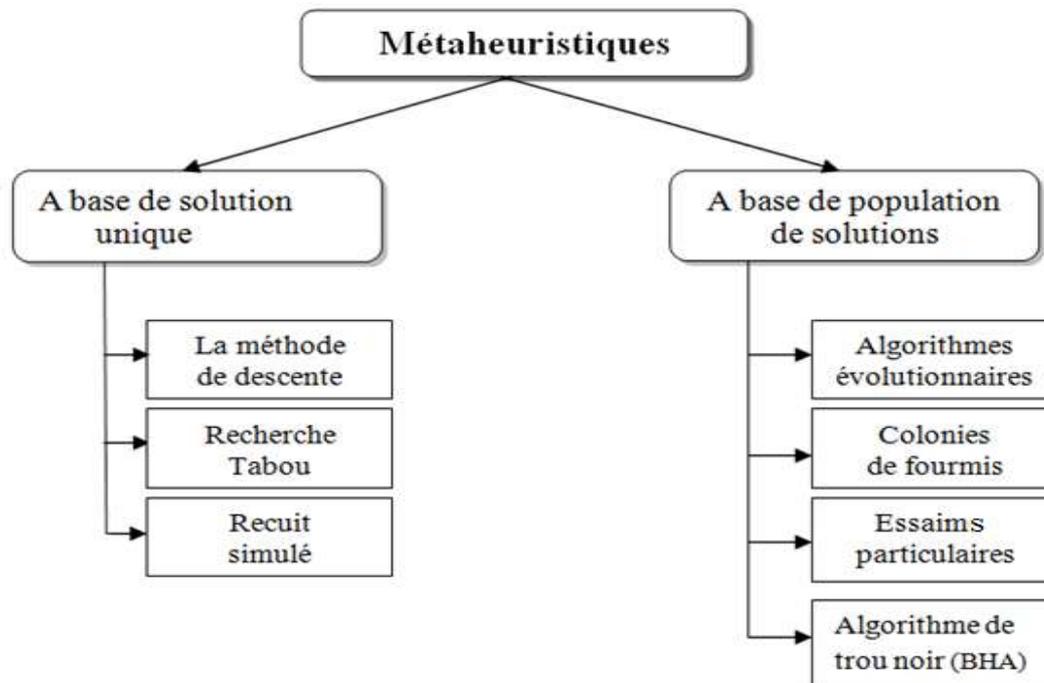


Figure I.6: Classification des métaheuristiques.

I.6. Exploration et exploitation des algorithmes d'optimisation

Les deux termes exploitation et exploration sont appelées aussi «intensification» et «diversification». Pour un algorithme d'optimisation, l'exploration est sa capacité à explorer le domaine des variables pour rechercher la meilleure vallée, c'est à dire celle qui contient l'optimum global. A l'inverse, l'exploitation est sa capacité à converger rapidement vers le minimum dans la plupart du temps d'un compromis entre l'exploration et l'exploitation. Certaines méthodes toute fois n'utilisent qu'un seul de ces opérateurs pour parvenir à l'optimum. Ainsi, les méthodes déterministes, exploitant les dérivées de la fonction objective et des contraintes pour atteindre rapidement et précisément le minimum local le plus proche du point de départ, privilégient l'exploitation au détriment de l'exploration. Tout algorithme d'optimisation doit utiliser ces deux stratégies pour trouver l'optimum global : l'exploration pour la recherche de régions inexplorées de l'espace de recherche et l'exploitation pour exploiter la connaissance acquise aux points déjà visités et ainsi trouver des points meilleurs. [14]

I.7. Conclusion

Dans ce chapitre, nous avons présenté un aperçu sur les métaheuristicques les plus populaires pour l'optimisation combinatoire. Tandis que l'optimisation combinatoire peut être classée en deux méthodes, *les méthodes exactes* et *les méthodes approchées*. Où les méthodes exactes permettent d'aboutir à la solution optimale, mais elles sont trop mauvais en termes de temps de calcul requis. Cependant, les méthodes approchées demandent des coûts de recherche raisonnables. Mais, elles ne garantissent pas l'optimalité de la solution. Pour les méthodes métaheuristicques (les méthodes approchées) qui sont notre étude, nous avons trouvé que les méthodes métaheuristicques peuvent être partagées en deux sous classes: des méthodes à base d'une solution unique et des méthodes à base de population de solutions. Les méthodes de la première sous classe se basent sur la recherche locale pour trouver la solution du problème à traiter. Par contre, les méthodes de la deuxième classe se basent sur une recherche globale ce qui leur permet d'échapper au problème de la convergence vers l'optimum global et augmente leur possibilité de fournir des solutions de bonnes qualités.

Chapitre III

Application

Chapitre II

II.1. Introduction

L'utilisation des algorithmes évolutifs pour la résolution des problèmes combinatoires, est devenue un sujet très important et populaire dans de nombreux domaines scientifiques. Ils existent plusieurs algorithmes évolutifs inspirés de différents phénomènes naturels tels que l'algorithme génétique, l'algorithme d'essaims de particules etc.

Un nouvel algorithme inspiré de la nature et modélise les caractéristiques du phénomène de trou noir est proposé par Abdolreza Hatamlou. Ces caractéristiques se manifestent dans la convergence des stars vers le trou noir, et le déplacement de ce dernier dans l'espace. L'algorithme du trou noir est parmi les algorithmes qui ont attiré une attention significative pendant les dernières années pour résoudre les problèmes d'optimisation complexes et combinatoires [17]

Dans ce chapitre; premièrement, nous présenterons le phénomène de trou noir dans l'espace et ses caractéristiques, deuxièmement, nous détaillerons la méthode d'optimisation inspirée du trou noir, en présentant son algorithme et ses formulations mathématiques.

II.2. Phénomène du trou noir

Le premier à prédire l'existence des trous noirs dans le monde est "Albert Einstein" en 1916 dans sa théorie du "relativisme général". Le terme «trou noir» a été utilisé pour la première fois en 1969 par le scientifique américain John Wheeler. Le premier trou noir a été découvert en 1971. [18]

Le trou noir est une zone dans l'espace où la force des nuages de gravité est assez grande que la lumière elle-même ne peut pas s'échapper. la raison de cette puissante d'attraction est la compression et la contraction du matériaux dans un petit espace.

Les trous noirs ne mouvement pas aléatoirement dans l'espace, mais soumis aux lois de la gravité comme d'autres objets dans l'espace.

Le trou noir peut être déterminé en faisant tourner les stars autour d'une zone spécifique de l'espace.



Figure II.1 : Vue schématique d'un trou noir

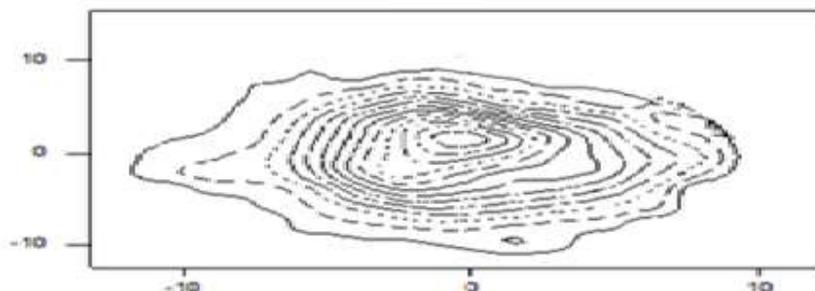


Figure II.2 : Le trou noir et la distribution des stars dans l'espace

Autour d'un trou noir, il y a une surface mathématiquement définie appelée un horizon d'événements qui marque le point de non-retour. A ce rayon, la vitesse d'échappement est égal à la vitesse de la lumière, Si quelque chose se rapproche de l'horizon de l'événement ou traverse le rayon de Schwarzschild (horizon d'événements), il sera absorbé dans le trou noir et disparaîtra définitivement. On l'appelle "noir" parce qu'il absorbe toute la lumière qui frappe l'horizon et ne reflétant rien. [19]

II.3. Algorithme BHA

L'algorithme trou noir (en anglais black hole algorithm) est un algorithme évolutif inspiré de phénomène trou noir(BHA) pour résoudre les problèmes combinatoires complexes et fournir des solutions optimales dans un temps raisonnable. Dans cette section nous allons présenter le principe de base de cet algorithme et son organigramme. [19]

II.3.1. Description générale de BHA

Comme tous les métaheuristiques, la méthode BHA est basée sur une population des solutions candidates désignée pour résoudre un problème donné. La population de solutions candidates sont distribuée, initialement, au hasard dans l'espace de recherche, puis certains mécanismes sont utilisé pour converger itérativement vers la solution optimale. Tant que l'algorithme BHA est inspirée du phénomène de trou noir(BHA), ses étapes suivent le même mécanisme de ce phénomène

Dans cette méthode, l'algorithme BHA commence avec une population initiale de solutions candidates à un problème d'optimisation et une fonction objective qui leur est calculée. A chaque itération de l'algorithme BHA, le meilleur candidat est sélectionné pour être le trou noir, qui commence alors à en tirer d'autres candidats, appelés les stars. si un star se rapproche trop du trou noir(BHA), elle sera avalée par le trou noir(BHA) et disparaîtra pour toujours. Dans ce cas, une nouvelle star (solution candidate) est générée aléatoirement et placée dans l'espace de recherche et lance une nouvelle recherche. [19]

II.3.2. Mécanismes de fonctionnement de la méthode BHA

A partir du principe général de l'algorithme de trou noir, on peut résumer les différentes étapes de cet algorithme comme suit :

- 1) l'initialisation
- 2) Evaluation.
- 3) Sélection la meilleur solution trouvé.
- 4) Mise à jour (changer les positions).
- 5) L'événement de l'horizon du trou noir.[19]

a). Initialisation de la population

Comme dans tout problème d'optimisation, une connaissance de "bons" points de départ conditionne la rapidité de la convergence vers l'optimum. Si la position de l'optimum dans l'espace de recherche est totalement inconnue, il est naturel de générer aléatoirement des individus en faisant des tirages uniformes dans chacun des domaines associés aux composantes de l'espace de recherche, en veillant à ce que les individus produits respectent les contraintes [19].

Initialiser une population des stars avec des emplacements aléatoires dans l'espace de recherche peut être définir comme suit :

$$P(t) = \{x_1^t + x_2^t + x_3^t + x_4^t \dots + x_n^t\}$$

P est la population initiale, où chaque solution est définie comme suit :

$$x_i^t = l_{min} + (l_{max} - l_{min}) \times rand$$

l_{max} et l_{min} définissent la borne supérieure et inférieure de l'espace de recherche.

b). Evaluation

Pour recherche de l'individu le plus adapté selon les critères souhaites .les solutions candidates seront évalué et représentée par leur valeur individu de fitness.

$$f_i = \sum_{i=1}^{\text{taille_pop}} \text{eval}(p(t))$$

Exemple : Soit la fonction $f = x^2$ est la fonction de fitness, n est le nombre maximal des solutions candidates.

Solution	Les valeurs de fitness
$X_1 = 0.1$	$f(x) = (0.1)^2 = 0.01$
$X_2 = 0.2$	$f(x) = (0.2)^2 = 0.04$
$X_3 = 0.3$	$f(x) = (0.3)^2 = 0.09$
.	.
.	.
.	.
.	.
X_n	$f(x) = (x_n)^2$

c). Sélection

Durant l'étape de sélection, le meilleur individu de la population, qui a la valeur optimal de fitness, est considéré comme le trou noir et les individus qui reste forment les stars normales.

$$x_{BH} = \arg \min_{x_i} f(x_i), \quad 1 \leq i \leq N$$

d). Mise à jour (changement des positions)

Durant la mise à jour, une nouvelle population est générée, en identifiant une nouvelle position de chaque star selon l'équation (2)

$$x_i(t+1) = x_i(t) + rand * (x_{BH} - x_i(t)) \quad i = 1, 2, \dots, N \quad (2)$$

Cette formulation traduit le phénomène d'absorption des stars par le trou noir. Le trou noir x_{BH} absorbe les stars qui l'entourent et toutes les stars commencent à se déplacer vers lui.

Où

- ✓ $x_i(t)$ et $x_i(t+1)$: les localisations de star à chaque itération t et $t+1$, respectivement.
- ✓ x_{BH} : l'emplacement du trou noir dans l'espace de recherche.
- ✓ $rand$ est un nombre aléatoire dans l'intervalle $[0, 1]$.
- ✓ N est le nombre des stars (solutions candidates).

Il faut noter que, une star $x_i(t)$ peut atteindre un endroit, avec une valeur de fitness meilleure que celle du trou noir. Dans ce cas, ceci le trou noir est remplacé par $x_i(t)$; $x_{BH} = x_i(t)$. [19]

g). l'horizon des événements du trou noir

De plus, il y a la probabilité de traverser l'horizon des événements pendant le déplacement des stars vers le trou noir. Chaque star (solution candidate) qui traverse l'horizon des événements du trou noir sera aspirée par le trou noir. Dans ce cas, cette star meurt et une autre solution candidate (star) est née et distribuée au hasard dans l'espace de recherche et commence une nouvelle recherche. Ceci est fait pour garder le nombre de solutions candidates constantes. [19]

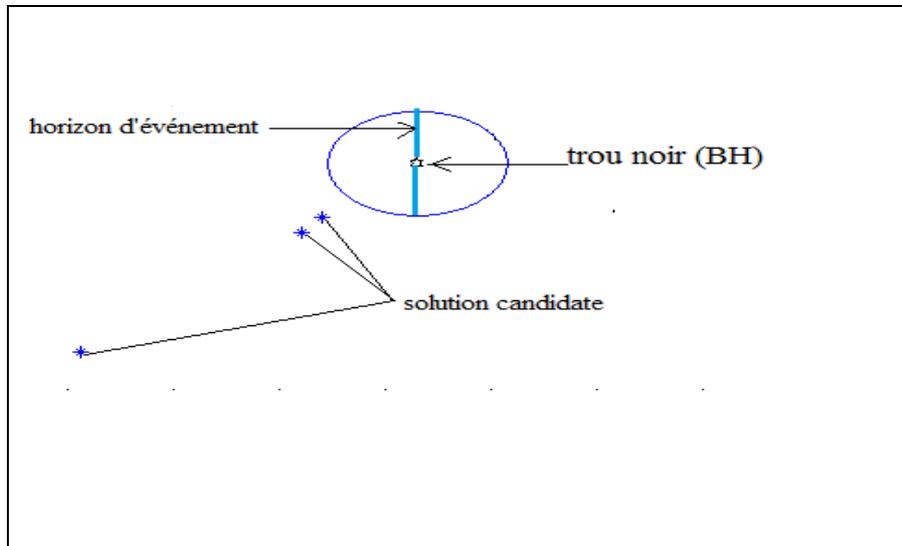


Figure. II.3. : Vue les éléments d'horizon des événements.

Le rayon de l'horizon des événements dans l'algorithme du trou noir est calculé en utilisant l'équation suivante :

$$R = \frac{f_{BH}}{\sum_{i=1}^N f_i} \quad (3)$$

- ✓ f_{BH} : la valeur de fitness du trou noir.
- ✓ f_i : la valeur de fitness de chaque star.
- ✓ N est le nombre les stars (solutions candidates)

Lorsque la distance entre une solution candidate et le trou noir (meilleur candidat) est inférieure à R, ce candidat est détruit et un nouveau candidat est créé et distribué de manière aléatoire dans l'espace de recherche.

f). Critère d'arrêt.

Les étapes de l'algorithme trou noir se répètent durant un nombre d'itération. Si un critère de terminaison est satisfait, le processus se termine. Généralement les critères d'arrêt utilisés sont le nombre maximal d'itérations et la valeur de fitness obtenu est suffisamment bonne.[19]

Algorithme BHA

```

Début
Initialiser une population de N points
Pour chaque i faire
{
  Applique la fonction de fitness
}
Sélection de la meilleure solution (BH)
t=(nombre de itération )
Pour i=1 à t faire {
  Pour j=1 à N faire {
    Générer les solutions basées sur l'équation Eq.2
  }
  Si (fitness de  $x_i(t+1)$  est meilleure que la fitness de  $x_i(t)$ ) {
    remplacer le star ancien de  $x_i(t)$  à nouveau star de  $x_i(t+1)$ 
  }
  Si non (c.-à-d. le fitness de  $x_i(t+1)$  est meilleure que la fitness de  $X_{BH}$ ) {
    remplacer  $X_{BH}$  à nouveau star de  $x_i(t+1)$ 
  }
  Calculer le rayon R en utilisant Eq.3

  Si (une solution travers le rayon R) {
    Générer aléatoirement une nouvelle solution (star)

    Si (le critère d'arrêt est atteint) {
      La meilleure solution est  $X_{BH}$ 
    }
    Si non {
      Rénover les solutions basées sur l'équation Eq.2
    }}
  Si non {
    Si (le critère d'arrêt est atteint) {
      La meilleure solution est  $X_{BH}$ 
    }
    Si non {
      Rénover les solutions basées sur l'équation Eq.2
    }}}
Fin

```

Figure. II.4. : Algorithme du trou noir (BHA)

II.4. Avantages et Inconvénients de l'algorithme BHA

La méthode de BHA possède quelques avantages et inconvénients:

a) Les avantages

- ✚ Il a une structure simple et il est facile à mettre en œuvre.
- ✚ Il n'a pas beaucoup de paramètres à régler.

b) Les inconvénients

- ✚ La possibilité d'éviter la solution optimale pendant la recherche dans certains problèmes.

II.5. Conclusion

Dans ce chapitre nous avons présenté une nouvelle méthode métaheuristique qui est l'optimisation inspirée de phénomène trou noir, Cette méthode de recherche basée sur la population, est inspirée des caractéristiques trous noirs dans la nature. Les avantages de cet algorithme est sa simplicité d'implémentation, et il n'a pas beaucoup de paramètres à régler et son inconvénient est qu'il peut éviter la solution optimale pendant la recherche.

Chapitre III

Application

Optimisation de problème de routage de véhicules

Chapitre III

III.1 Introduction

Dans le chapitre précédent nous avons présenté la théorie d'une nouvelle méthode d'optimisation inspirée à partir du phénomène de trou noir (BHA). Dans ce chapitre, nous allons l'appliquer sur quelques fonctions tests connues. Et puis nous envisageons de l'appliquer sur un problème réel qui est le problème de routage de véhicules.

Pour réaliser cette application on a utilisé une programmation sous Matlab (2014) sous Windows 7 sur un PC portable ACER équipé d'un processeur Intel® Core (TM) i3 -2328 CPU @ 2.20 GHz 2.20 GHz et une RAM de 4.00Go.

III.2. Organigramme de l'algorithme BHA :

Les concepts de l'algorithme BH présentés précédemment nous permettent de construire l'organigramme suivant :

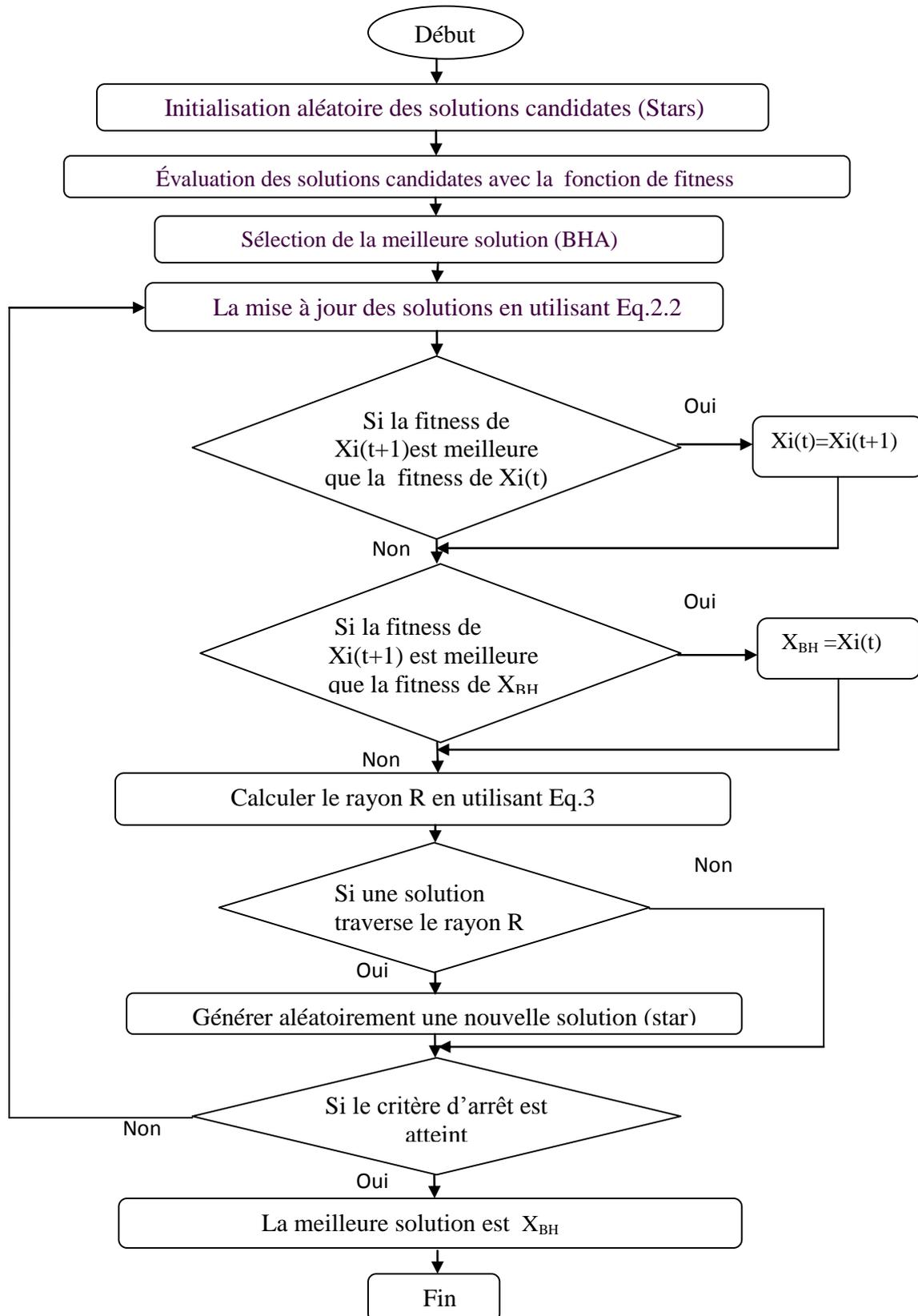


Figure III .1: Les étapes de l'algorithme du trou noir.

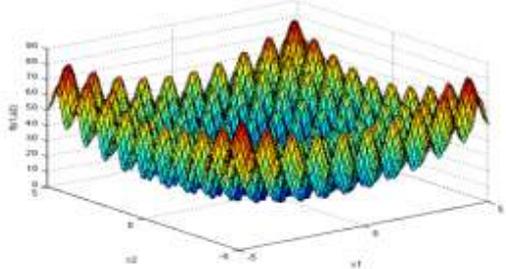
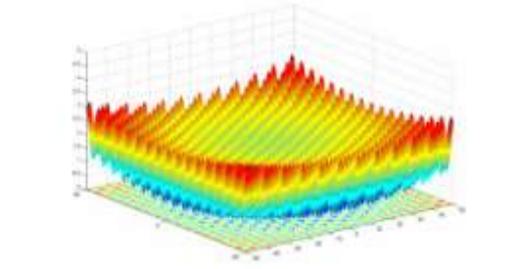
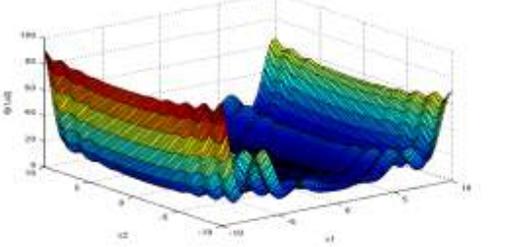
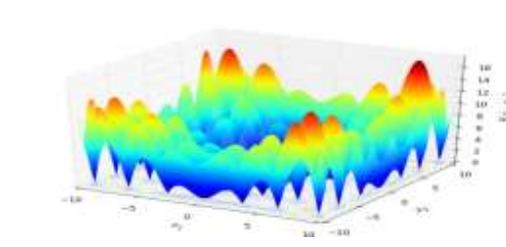
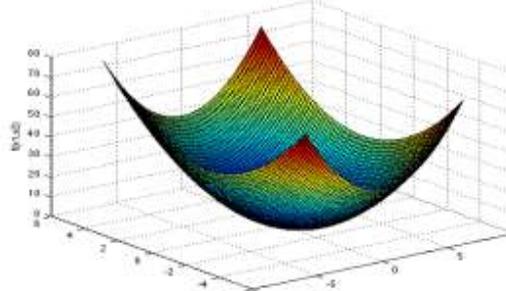
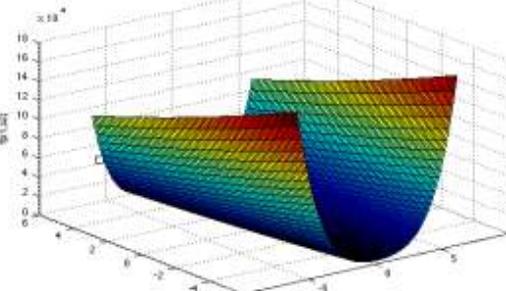
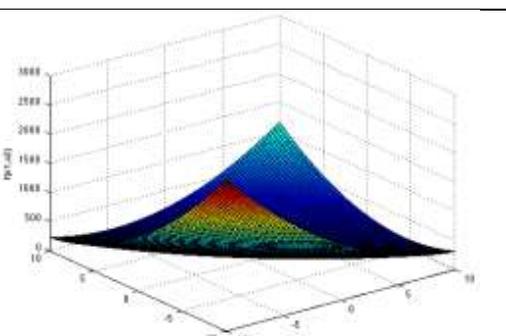
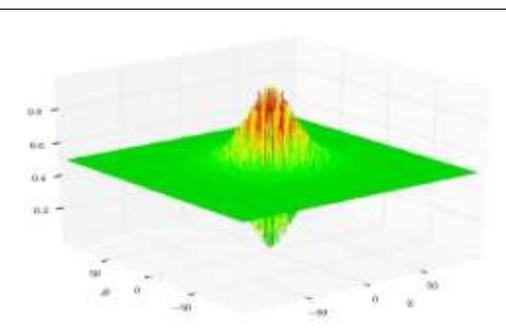
III.3. les fonctions de teste

Dans cette section, nous présentons quelques fonctions de tests, et leurs courbes en 3D.

Tableau III.1 : Les fonctions de test

	Nom de fonction	Formule de fonction	Espace de recherché
plusieurs minimums local	Alpine	$f(x) = \sum_{i=1}^d x_i \sin(x_i) + 0.1x_i $	[-10, 10]
	Griewank	$f(x) = \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600, 600]
	Levy	$f(x) = \sin^2(\pi\omega_1) + \sum_{i=1}^d (\omega_i - 1)^2 [1 + 10\sin^2(\pi\omega_d + 1)] + (\omega_d - 1)^2 [1 + \sin^2(2\pi\omega_d)]$ Where $\omega_i = 1 + \frac{x_i - 1}{4}$	[-10, 10]
	Rastrigin	$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10\cos(2\pi x_i)]$	[-5.12, 5.12]
	Schaffer2	$f(x) = 0.5 + \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{\left[1 + 0.001(x_1^2 + x_2^2)^2\right]^2}$	[-100, 100]
En forme sphérique	Sphere	$f(x) = \sum_{i=1}^d x_i^2$	[-5.12, 5.12]
En forme de la vallée	Rosenbrock	$f(x) = \sum_{i=1}^{d-1} \left[100(x_{i+1} - x_i)^2 + (x_i - 1)^2 \right]$	[-2.048, 2.048]
En forme de plaque	Booth	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	[-10, 10]

Tableau III.2 : Les courbes en 3D des fonctions de tests

 <p>fonction Alpine</p>	 <p>fonction Griewank</p>
 <p>fonction Levy</p>	 <p>fonction Rastrigin</p>
 <p>fonction Sphere</p>	 <p>fonction Rosenbrock</p>
 <p>fonction Booth</p>	 <p>fonction schaffer2</p>

III.4 Les résultats de teste de l'algorithme BHA

Dans cette section, nous avons appliqués quelques fonctions de teste, et les résultats de son application, pour découvrir certaines propriétés affectant l'algorithme, ensuite, en vue de comparaison, nous avons appliqué trois méthodes d'optimisations sur des fonctions de teste. Ces méthodes sont la méthode de recuit simulé (SA) et la méthode d'optimisations par essaims particulaires (PSO) et notre méthode (BHA), pour valider cette technique,

III.4.1 Les résultats de teste de l'algorithme BHA sur les fonctions tests

Nous allons tester la méthode BHA sur quelques unes de ces fonctions en variant à chaque fois un paramètre.

A) L'espace de recherche :

Les tableaux suivants présentent les résultats de teste des fonctions par la méthode BH en variant l'espace de recherche

Tableau III.3 : Résultats de teste des fonctions par BHA pour l'espace de recherche $[-10,10]$, et la solution optimale 2D.

Nom de fonction	Dimension	Nbr d'itération	Solution optimale	Fitness
Alpine	2	50	[-0.1002 -0.0000]	3.6086e-17
Griewank	2	50	[0.0130 -0.0071]	9.6482e-05
Levy	2	50	[1.0000 1.0000]	7.3849e-25
Rastrigin	2	50	1.0e-07 * [0.0595 -0.1397]	4.6185e-14
Sphere	2	50	1.0e-13* [0.5022 - 0.6466]	6.7035e-27
Rosenbrock	2	50	[0.9975 0.9950]	6.2912e-06
Booth	2	50	[1.0000 3.0000]	0
Schaffer2	2	50	1.0e-06 * [0.0928 -0.2150]	0

Tableau III.4 : résultats de testé les fonctions par BHA pour l'espace de recherche [-100, 100] et la solution optimale 2D.

Nom de fonction	Dimension	Nbr d'itération	Solution optimale	Fitness
Apline	2	50	[-0.1314 3.2418]	0.0041
Griewank	2	50	[3.1400 -4.4384]	0.0074
Levy	2	50	[1.0000 1.0000]	9.1043e-13
Rastrigin	2	50	1.0e-09 * [-0.3605 0.3450]	0
Sphere	2	50	1.0e-16 * [-0.0208 -0.2521]	6.4011e-34
Rosenbrock	2	50	[-0.5843 0.2825]	2.8567
Booth	2	50	[1.0000 3.0000]	1.9722e-29
schaffer2	2	50	1.0e-06 * [0.2299 0.0403]	0

Tableau III.5 : résultats de testé les fonctions par BHA pour l'espace de recherche [-1000, 1000] et la solution optimale 2D.

Nom de fonction	Dimension	Nbr d'itération	Solution optimale	Fitness
Apline	2	50	[-15.6078 9.5249]	3.5296e-09
Griewank	2	50	1.0e-04 * [-0.2601 0.4559]	8.5848e-10
Levy	2	50	[1.0000 1.0000]	1.4998e-32
Rastrigin	2	50	[-0.0000 -0.9950]	0.9950
Sphere	2	50	1.0e-07 * [0.0475 0.1225]	1.7260e-16
Rosenbrock	2	50	[-6.0701 36.8338]	50.0032
Booth	2	50	[1,0000 3,0000]	8.1372e-23
Schaffer2	2	50	1.0e-06 * [-0.2529 0.0934]	0

À travers ces tableaux ci-dessus, Nous remarquons à chaque augmentation dans l'espace de recherche, les valeurs de fitness varient; Donc les solutions ne sont pas bonnes pour certaines fonctions.

B) Dimensions et Itérations

Nous avons choisi quelques fonctions pour appliquer la métaheuristique. Pour voir son comportement dans les situations où on varie la dimension et le nombre des itérations tout en fixant l'espace de recherche.

Les deux tableaux suivant présentent les résultats de teste des fonctions par l'algorithme BHA dans le cas de variation de ces paramètres de l'algorithme où nous avons fait fixé le nombre des itérations et varié la dimension

Tableau III.6 : résultats de teste des fonctions par BHA et la solution optimale 2D.

Nbr d'itération=20				
Nom de fonction	Espace de Recherche	Dimension	Solution optimale	Fitness
Levy	[-10 10]	2	[1.0000 1.0000]	3.3017e-15
Booth	[-10 10]	2	[1.0000 3.0000]	1.7826e-16

Tableau III.7: résultats de testé les fonctions par méthode BHA et la solution optimale 5D.

Nbr d'itération=20				
Nom de fonction	Espace de Recherche	Dimension	Solution optimale	Fitness
Levy	[-10 10]	5	[1.0331 0.9589 1.0808 -0.0619 0.7382]	0.1003
Booth	[-10 10]	5	[1.0000 3.0000 2.0331 -3.3985 -2.2451]	1.3497e-10

Fonction Levy :

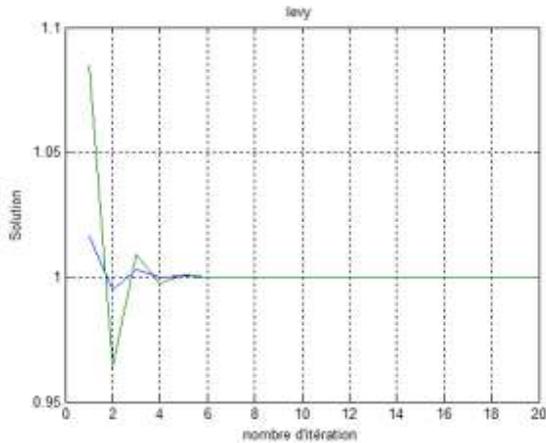


Figure III.2: variation de la solution optimale en fonction d'itérations (2D).

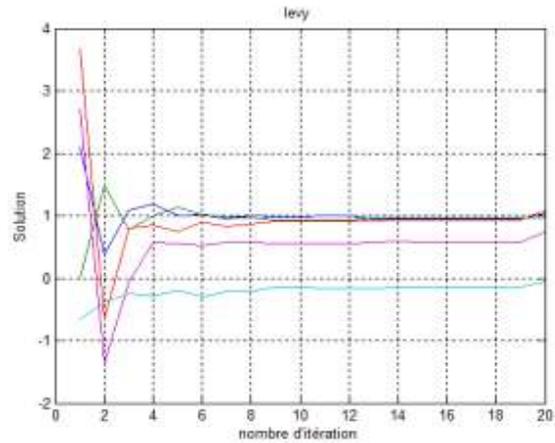


Figure III.3 variation de la solution optimale en fonction d'itérations (5D).

Les figures 8 et 9 présentent la variation de la solution optimale en fonction des itérations. Dans le cas de 2D nous remarquons que la solution optimale converge vers la solution exacte à la 5^{ème} itération mais dans le cas de 5D la solution convergent vers une valeur approximative de l'optimale.

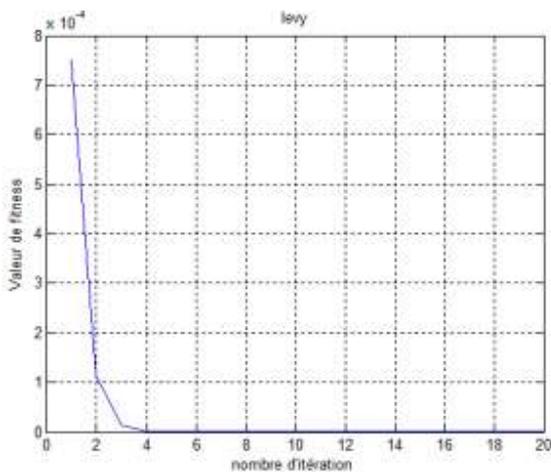


Figure III.4: variation de fitness en fonction d'itérations (2dimensions)

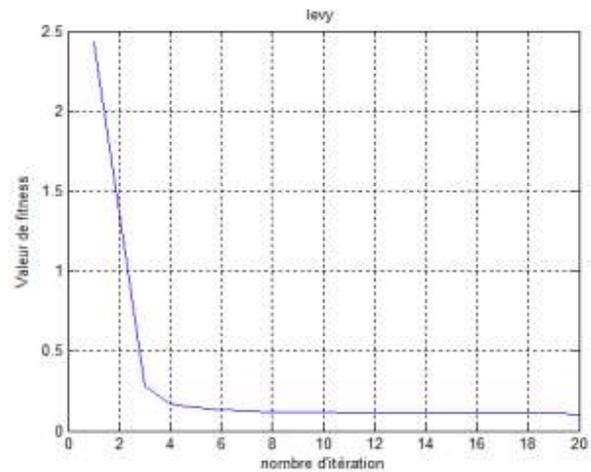


Figure III.5: variation de fitness en fonction d'itérations (5 dimensions)

Les figures 10 et 11 présentent la variation de fitness en fonction des itérations. dans le cas de 2D nous remarquons que la fonction de fitness diminue et converge vers la valeur zéro à la 4^{ème} itération. Cependant, dans le cas de 5D diminue mais ne converge pas vers la valeur zéro.

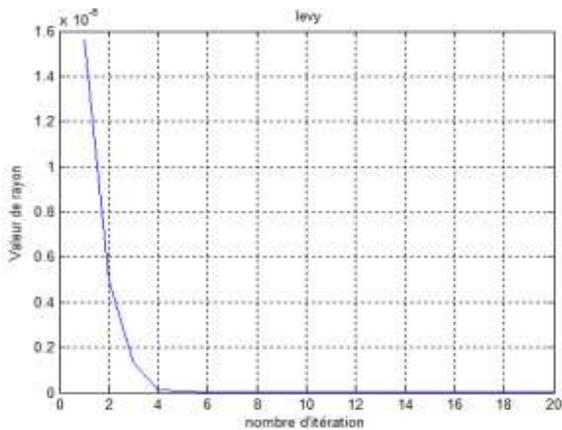


Figure III.6: variation de rayon en fonction d'itérations (2D).

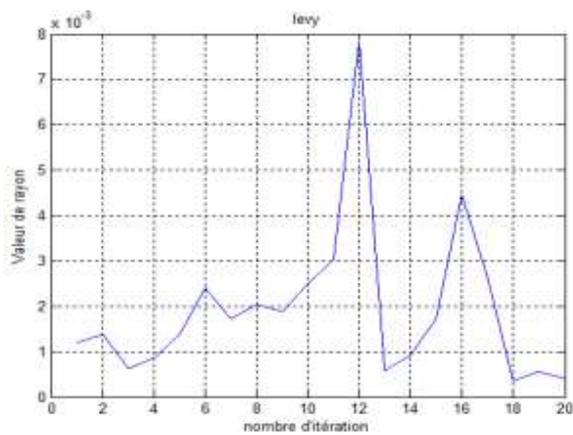


Figure III.7: variation de rayon en fonction d'itérations (5D).

La figure 6 et 7 présente la variation de la valeur de rayon en fonction des itérations. Nous remarquons que la valeur de rayon diminue et converge rapidement vers la valeur zéro à la 4^{ème} itération dans le cas de 2D, et dans le cas de 5D est de l'ordre de 10^{-3} mais instable.

Fonction Booth :

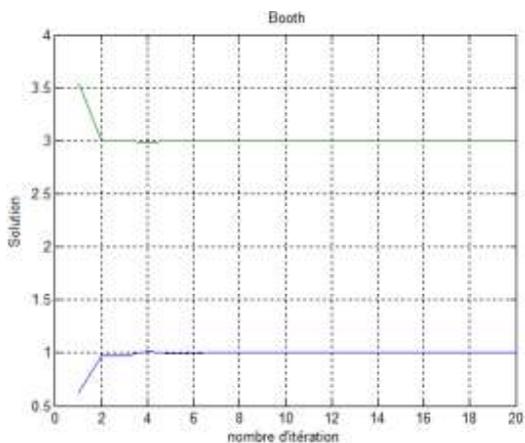


Figure III.8: variation de la solution optimale en fonction d'itération (2D).

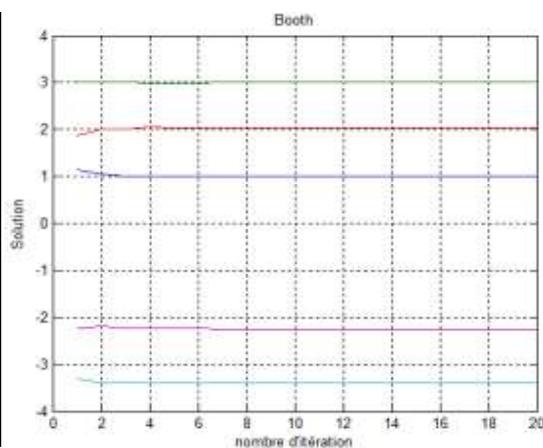


Figure III.9: variation de la solution optimale en fonction d'itération (5D)

Les figures 8 et 9 présentent la variation de la solution optimale en fonction des itérations. Dans le cas de 2D nous remarquons que la solution optimale converge vers la solution exacte dès la 2^{ème} itération mais dans le cas de 5D deux solutions sont converge vers la valeur optimal.

Fonction de Booth travaille en deux dimensions $x_1=1$ et $x_2=3$.

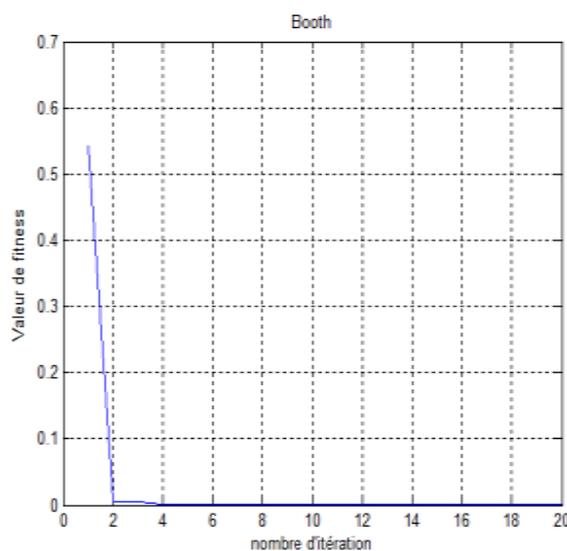


Figure III.10: variation de fitness en fonction d'itérations (2D)

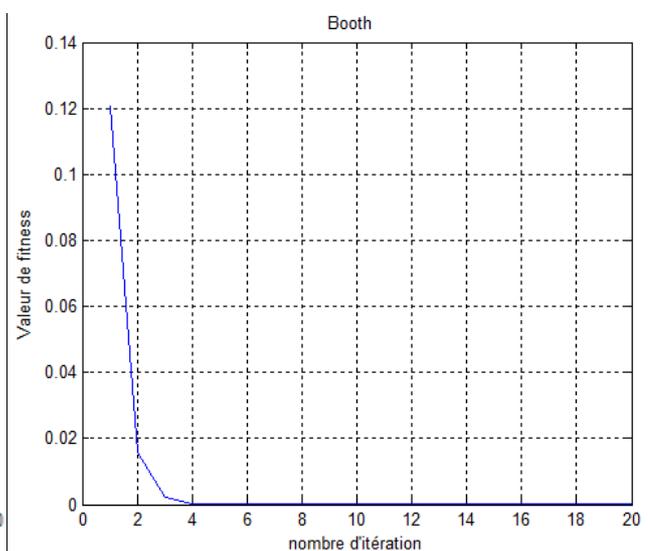


Figure III.11: variation de fitness en fonction d'itérations (5D)

Dans le cas de 2D nous remarquons la fitness en fonction d'itération, la fonction de fitness diminue et converge vers la valeur optimale à la 2^{ème} itération et dans le cas de 5D la fonction de fitness diminue et converge vers la valeur optimale à la 4^{ème} itération

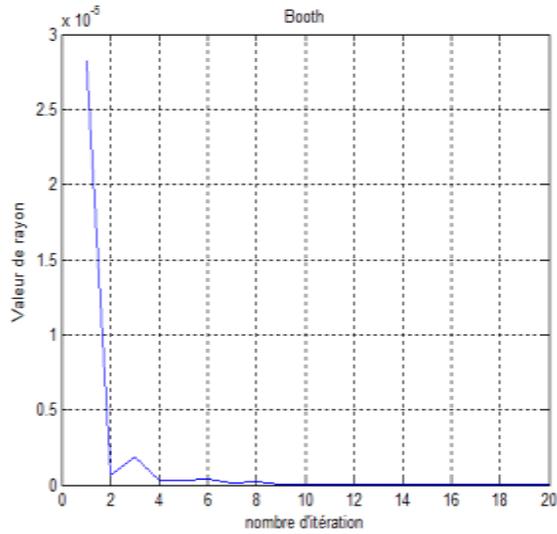


Figure III.12: variation de rayon en fonction d'itérations (2D).

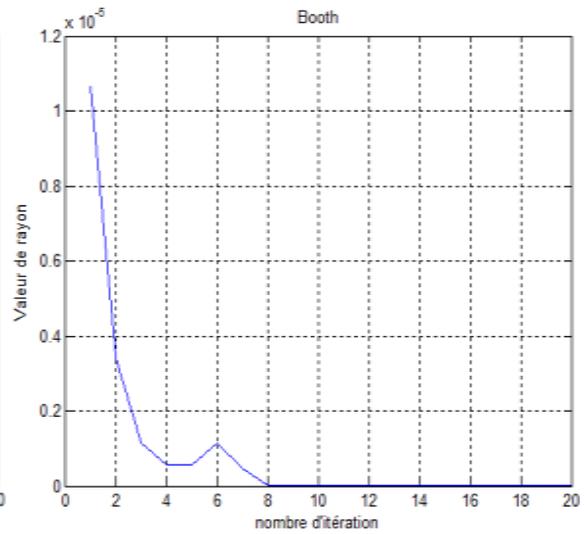


Figure III.13: variation de fitness en fonction d'itérations (5D).

La figure 12 et 13 présente la variation de la valeur de rayon en fonction des itérations. Nous remarquons que la valeur de rayon diminue et converge rapidement vers la valeur zéro dans le cas de 2D par rapport au cas de 5D

Les deux tableaux suivants expliquent les résultats de tests des fonctions par méthode BHA avec quelques paramètres de l'algorithme où nous avons fait fixer les paramètres. Hormis le nombre d'itération :

Tableau III.8: résultats de tests des fonctions par méthode BHA et la solution optimale 2D.

Nbr d'itération=20				
Nom de fonction	Espace de Recherche	Dimension	Solution optimale	Fitness
Rosenbrock	[-10 10]	2	[0.7240 0.5217]	0.0768
Schaffer2	[-10 10]	2	1.0e-06 * [-0.2002 0.1491]	0

Tableau III.9: résultats de tests des fonctions par méthode BHA et la solution optimale 2D.

Nbr d'itération = 50				
Nom de fonction	Espace de Recherche	Dimension	Solution optimale	Fitness
Rosenbrock	[-10 10]	2	[0.9996 0.9995]	1.1376e-05
Schaffer2	[-10 10]	2	1.0e-06 * [-0.2962 0.1009]	0

Fonction Rosenbrock :

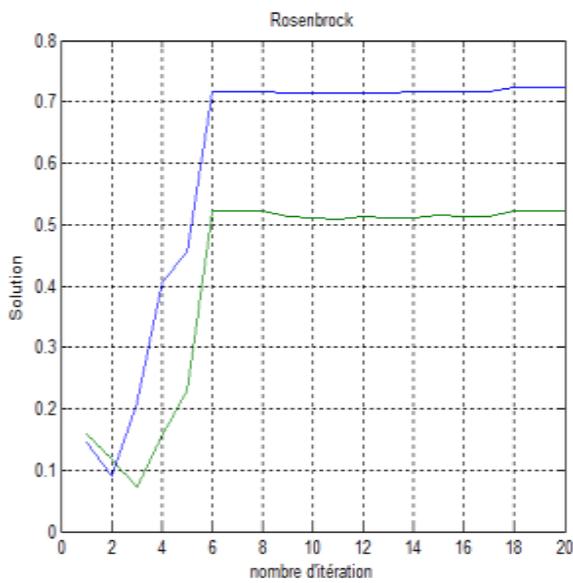


Figure III.14: variation de la solution optimale en fonction d'itérations (20 itérations).

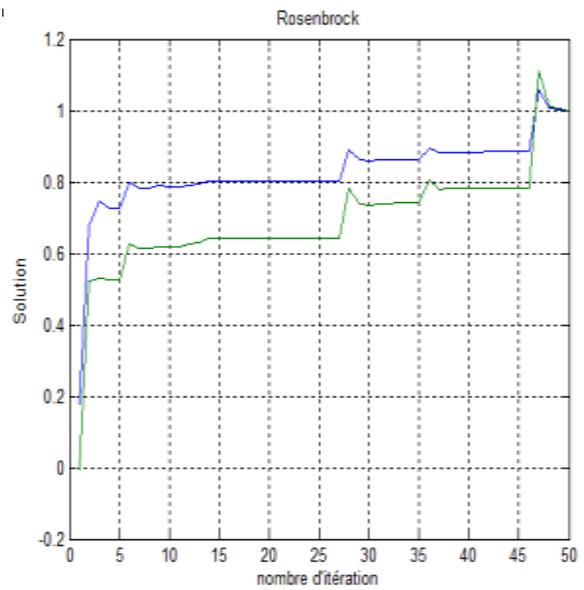


Figure III.15: variation de la solution optimale en fonction d'itérations (50 itérations).

Dans le cas de 20 itérations nous remarquons que les solutions sont variées jusqu'à l'itération 6 elles sont, ainsi, fixées. dans un domaine de $[0.52, 0.72]$ à peu près égal à un . Cependant, dans le cas de 50 itérations les solutions sont converges vers la valeur optimale.

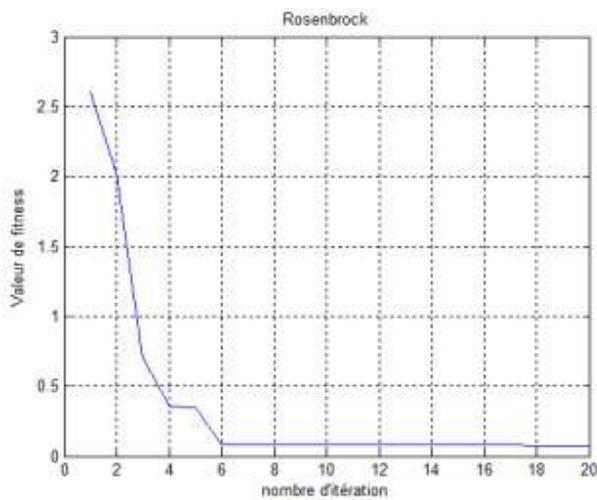


Figure III.16: variation de fitness en fonction d'itérations (20 itérations)

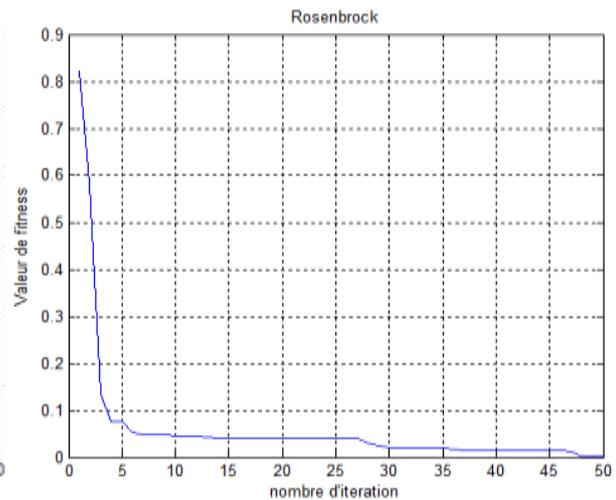


Figure III.17: variation de fitness en fonction d'itérations (50 itérations)

Dans le cas de 20 itérations nous remarquons la fonction de fitness diminue et converge vers la valeur optimale à la 6^{ème} itération . presque égal à 0.1 ainsi, fixées.

Cependant, dans le cas de 5D la fonction de fitness diminue et converge vers la valeur optimale à la 47^{ème} itération presque égal à 0.

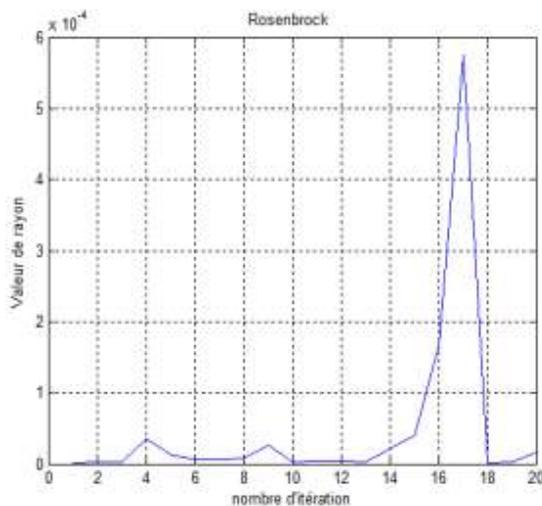


Figure III.18: variation de rayon en fonction d'itérations (20 itérations)

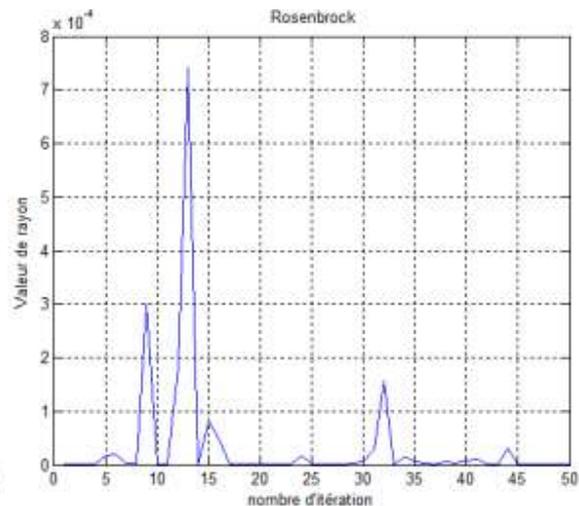


Figure III.19: variation de rayon en fonction d'itérations (50 itérations)

Nous remarquons que les valeurs des rayons sont instables; mais dans le cas de 5D où l'itérations est supérieur ;donc la valeur de rayon est converge vers zéro

Fonction Schaffer2:

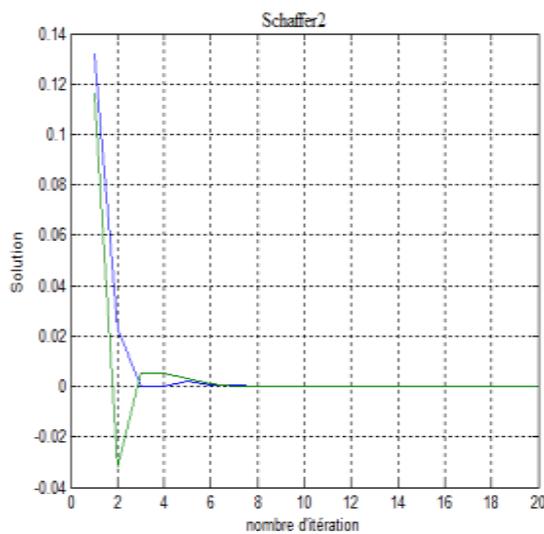


Figure III.20: variation de la solution optimale en fonction d'itérations (20 itérations).

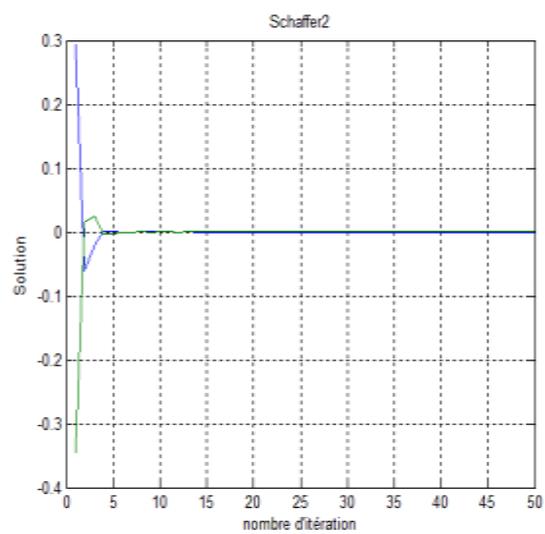


Figure III.21: variation de la solution optimale en fonction d'itérations (50 itérations).

Dans le cas de 20 itérations nous remarquons que la solution optimale converge vers la solution exacte à la 6^{ème} itération

Cependant, dans le cas de 50 itérations les solutions sont converge vers la valeur exacte à la 4^{ème} itération.

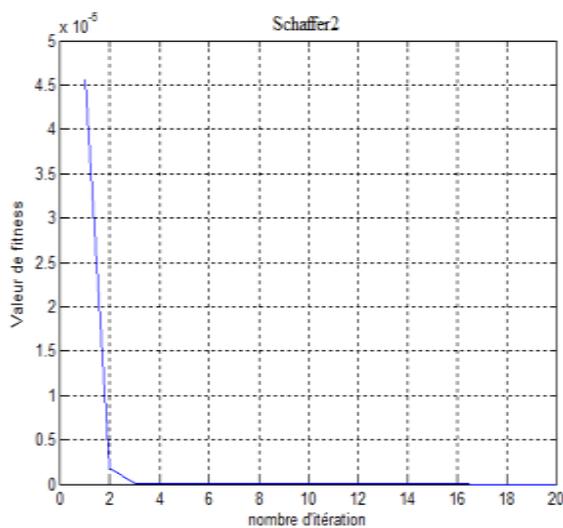


Figure III.22: variation de fitness en fonction d'itérations (20 itérations).

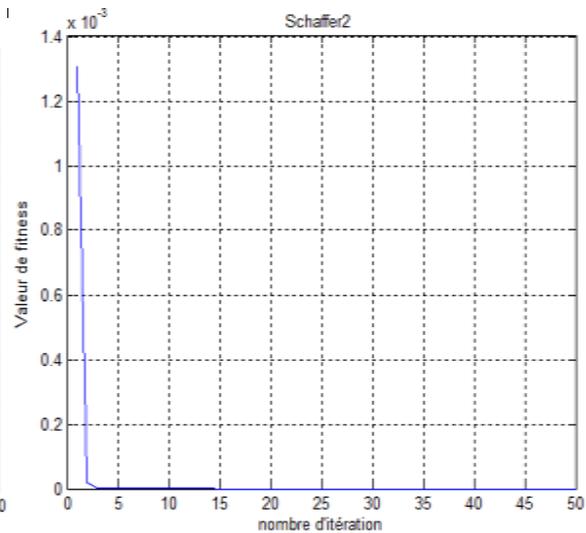


Figure III.23: variation de fitness en fonction d'itérations (50 itérations).

Dans les deux cas (20 itérations et 50 itérations) nous remarquons la fonction de fitness diminue et converge vers la valeur optimale à la 3^{ème} itération.

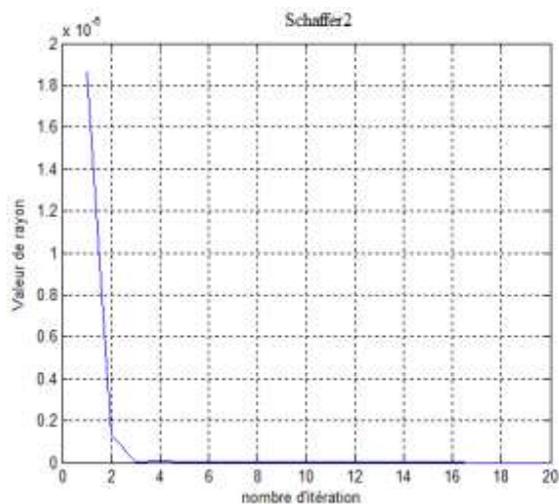


Figure III.24: variation de rayon en Fonction d'itérations (20 itérations)

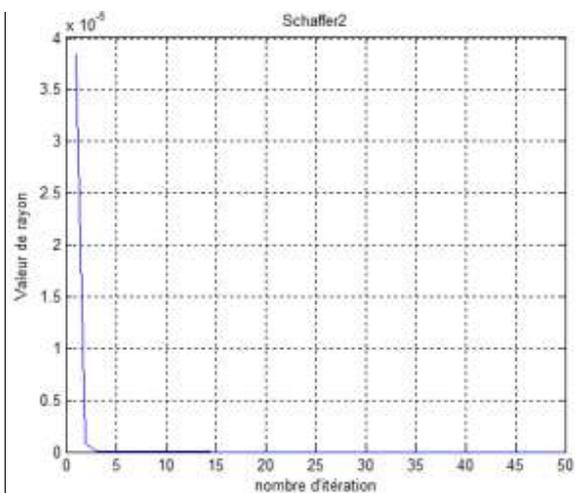


Figure III.25: variation de fitness en fonction d'itérations (50 itérations)

Nous remarquons que la valeur de rayon diminue et converge rapidement vers la valeur zéro dans le cas de 50 itérations par rapport au cas de 20 itérations .

III.4.2 Les résultats de teste des algorithmes BHA, SA et PSO sur les fonctions tests

Nous allons tester les méthodes BHA, SA et PSO sur les fonction de teste et comparer entre eux.

Tableau III.10: résultats les valeurs de fitness par des méthodes PSO et SA et BHA sur la fonction de tests dans l'espace de recherche [-10, 10], et 100 itérations

Méthode / Fonction	PSO	SA	BHA
Alpine	9.7073e-10	1.8123e-06	0
Booth	1.4268e-16	7.5120e-09	7.8886e-28
Sphere	2.1994e-18	3.8945e-11	0
Griewank	0	0.007396	0
levy	2.4399e-16	1.5498e-12	4.3529e-18
Rastrigin	6.0396e-14	3.3932e-08	0
Rosenbrock	1.9133e-09	8.9159e-06	1.5529e-13
Schaffer2	2.2204e-16	1.7097e-14	0

Tableau III.11: résultats des solutions optimale 2D des valeurs de fitness

Méthode / Fonction	PSO	SA	BHA
Alpine	[5.0339e-09 4.6734e-09]	[-7.1564e-06 1.0966e-05]	[-0.1002 0.1002]
Booth	[1.0000 3.0000]	[1.0000 3.0001]	[1.0000 3.0000]
Sphere	[-2.1774e-10 -1.4670e-09]	[-4.5459e-06 -4.2755e-06]	1.0e-28 [0.4880 0.0714]
Griewank	[-8.7484e-09 6.9601e-09]	[-3.1400 4.4384]	1.0e-07 * [-0.0434 0.1194]
Levy	[1.0000 1.0000]	[1.0000 1.0000]	[1.0000 1.0000]

Rastrigin	[1.2587e-08 1.1944e-08]	[1.3057e-05 7.4758e-07]	1.0e-08 * [-0.1103 -0.1493]
Rosenbrock	[1.0000 0.9999]	[0.9970 0.9940]	[1.0000 1.0000]
Schaffer2	[-2.5376e-07 -1.5166e-07]	[-3.1111e-06 2.7248e-06]	1.0e-06 * [-0.2810 0.0105]

On choisit une fonction alpine pour présenter la figure de résultats de fitness de trois méthodes

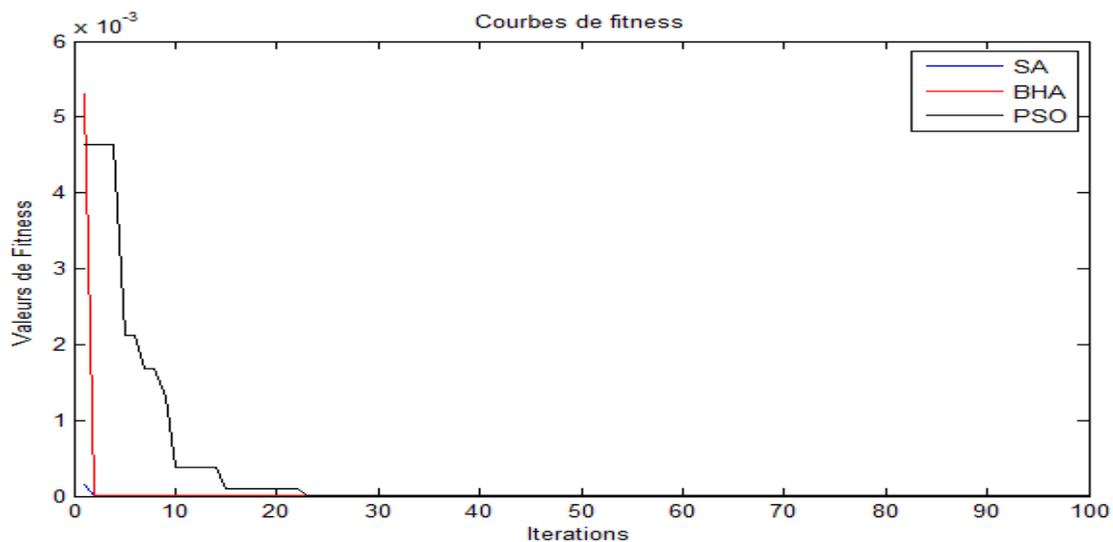


Figure III.26: Résultats de variation de fitness en fonction d’itérations (100 itérations) sur la fonction **Alpine** par les trois méthodes

Nous remarquons à travers les tableaux ci-dessus et la figure que les valeurs de fitness des fonctions de tests par l’algorithme BHA est convergée bien vers la valeur optimale zéro, par rapport des méthodes SA et PSO ; alors, nous remarquons que des solutions optimales de 2D à une fonction de bonne valeur.

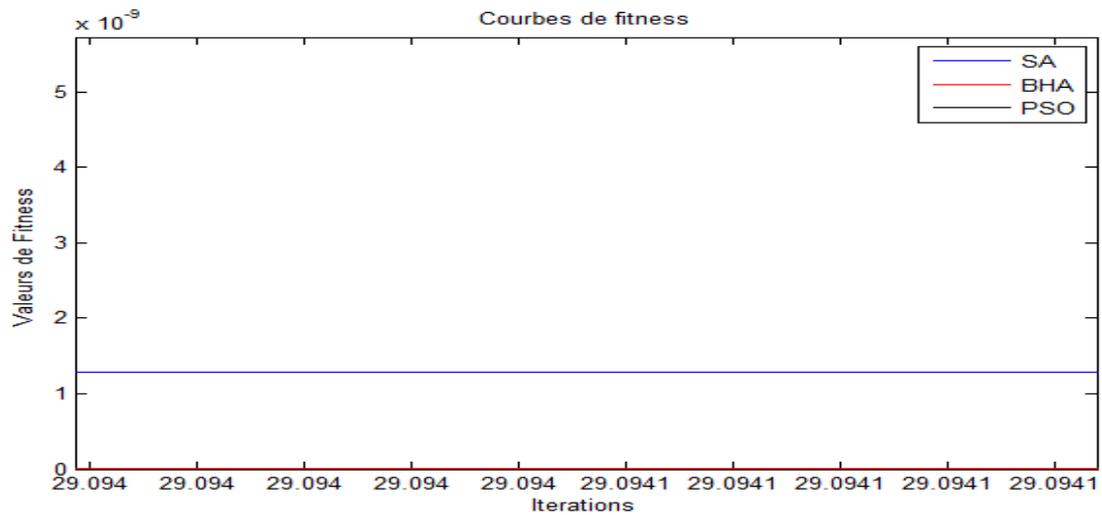


Figure III.27: la figure ci-dessus, pour éclairer les solutions optimales

Tableau III.12: Les fitness obtenues par les trois algorithmes [-10, 10], et 500 itérations

Méthode / Fonction	PSO	SA	BHA
Alpine	1.1102e-15	7.5239e-11	3.0474e-100
Booth	0	6.5616e-16	0
Sphere	5.5954e-258	3.4921e-17	1.0942e-112
Griewank	0	0	0
Levy	1.4998e-32	1.9496e-18	1.4998e-32
Rastrigin	0	3.5527e-15	0
Rosenbrock	0	1.3326e-12	1.446e-10
Schaffer2	0	0	0

Tableau III.13: résultats des solutions optimale 2D des fonctions de testes

Méthode / Fonction	PSO	SA	BHA
Alpine	1.0e-31 * [6.1830 3.1462]	1.0e-10 * [5.7348 -1.7890]	[-0.1002 0.0000]
Booth	[1.0000 3.0000]	[1.0000 3.0000]	[1.0000 3.0000]

Sphere	1.0e-129 * [1.2228-2.0249]	1.0e-09 * [-5.8563 -0.78990]	1.0e-56 * [0.4710 0.9340]
Griewank	1.0e-09 * [9.5391 -1.2091]	1.0e-09 * [7.7949 -1.6340]	1.0e-07 * [-0.0847 -0.1317]
Levy	[1.0000 1.0000]	[1.0000 1.0000]	[1.0000 1.0000]
Rastrigin	1.0e-09 * [0.44698 -1.6344]	1.0e-09 * [0.54251 -4.4404]	1.0e-10 * [-8,1644 -2,6793]
Rosenbrock	[1.0000 1.0000]	[1.0000 1.0000]	[1.0000 1.0000]
Schaffer2	1.0e-07 * [0.21554 3.1778]	1.0e-07 * [0.63851 1.8794]	1.0e-07 * [0.0314 -2,9949]

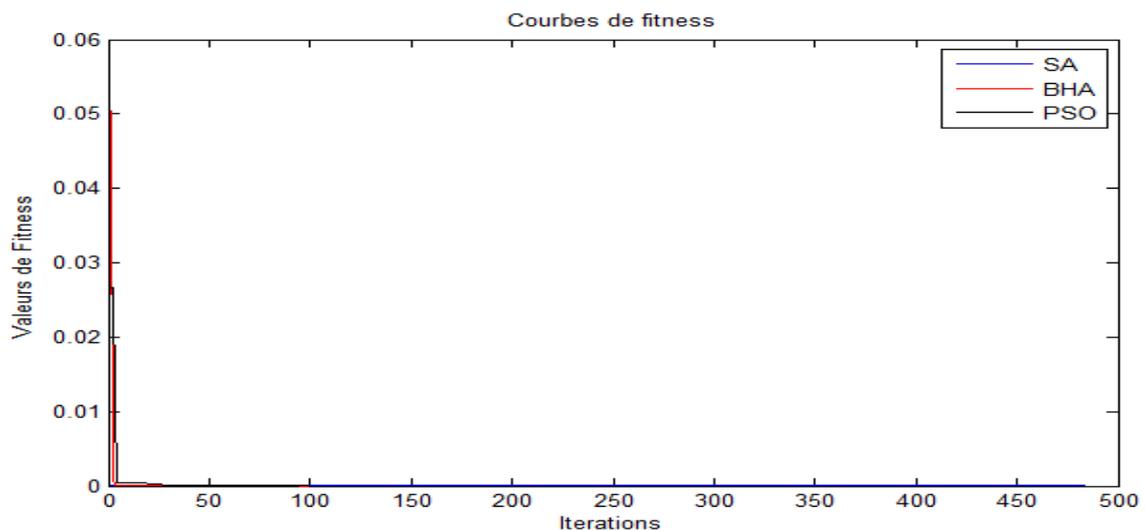


Figure III.28: Résultats de variation de fitness en fonction d'itérations (500 itérations) sur la fonction Alpine par les trois méthodes

Nous remarquons à travers les tableaux et la figure ci-dessus; quand on augmente le nombre itérations, les valeurs de fitness des fonction de tests par les algorithmes BHA et SA diminues et convergent bien et rapidement vers la valeur optimale zéro , comparer avec la méthode PSO; alors, nous remarquons que les solutions optimales à 2D de chaque fonction sont des bonnes valeurs .

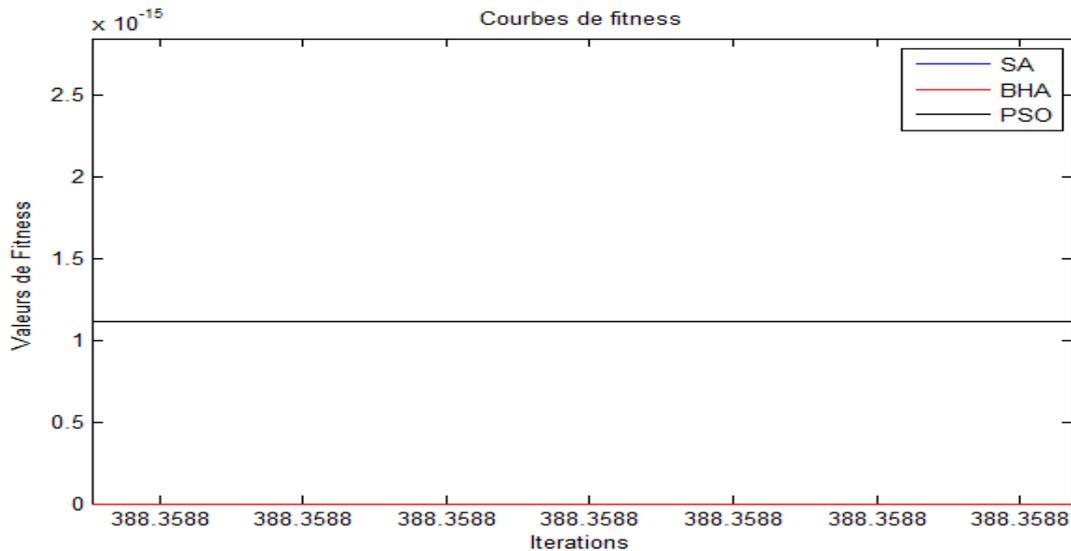


Figure III.29: la figure ci-dessus, pour éclairé les solutions optimales

Tableau III.14: résultats des valeurs de fitness par les méthodes PSO et SA et BHA sur des fonctions de tests dans l'espace de recherche [-10, 10], et 500 itérations et 5D

Méthode	PSO	SA	BHA
Alpine	1.6557e-16	6.5738e-07	6.0515e-06
Booth	0	8.2584e-16	0
Sphere	4.4072e-192	1.0459e-11	1.6019e-26
Griewank	0.017236	0.039417	0.007396
Levy	1.4998e-32	7.4092e-13	6.275e-09
Rastrigin	0	1.2115e-09	4.9748
Rosenbrock	0.03528	7.9795e-10	0.1805
Schaffer2	0	0	0

A partir des tableaux (14 et 12) ci-dessus; on voit que, quand on a augmenté la dimension à 5D, les valeurs de fitness des fonctions de tests obtenues par les trois méthodes, comparées avec celles obtenues en 2D **croissent** pour certaines fonctions; Donc les résultats pas bien .

III.5. Application sur le problème de routage de véhicules

III.5.1 Introduction

Le problème de routage de véhicule est fait partie des problèmes de transport, tout comme le problème du voyageur de commerce, il possède des composants de la vie réelle: route, clients, véhicules, centres de dépôt, conducteurs, contraintes et objectifs. Il est considéré de plus problèmes d'optimisation difficiles.[20]

Dans cette partie, nous commencerons par la définition le problème de routage de véhicule. Ensuite nous présenterons les différents modèles basé sur de ce problème, qui sera résolu par les méthodes d'optimisations métaheuristiques qui est SA, PSO et BHA sujet de notre étude et évaluer la performance de notre algorithme.

III.5.2 problème de routage de véhicules (VRP)

Le problème de routage de véhicules (aussi appelé VRP pour *Vehicle Routing Problem*) est une classe de problèmes d'optimisation combinatoire généralisés, il est une extension du problème du voyageur du commerce [Dhaenens et al., 2002]. Il a été introduit pour la première fois par Dantzig et al. [1954] sous le nom de *Truck Dispatching Problem* et a depuis fait l'objet d'études intensives pour le modéliser et le résoudre.[21]

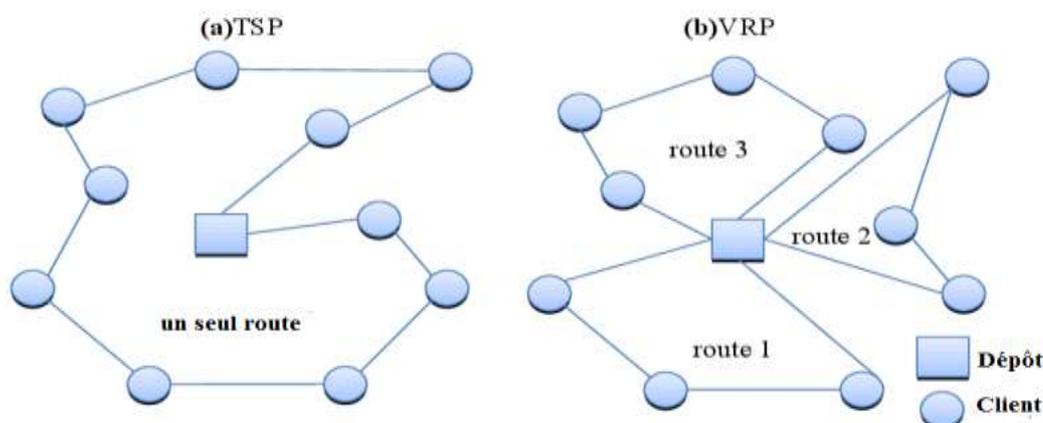


Figure III.30 : La différence entre le problème TSP et VRP

III.5.3 Description du problème de tournée de véhicule

La version la plus basique du VRP consiste à livrer des marchandises auprès des clients à l'aide d'une flotte de véhicules identiques à capacité limitée. Chaque client doit être desservi une et une seule fois et chaque tournée commence et se termine au dépôt. L'ensemble des clients visités par un véhicule désigne la tournée de celui-ci tel qui est représenté par la figure III.31

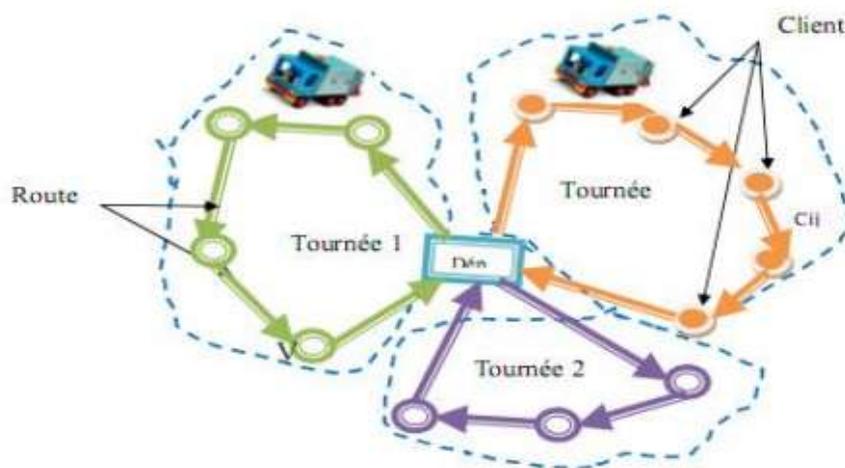


Figure III.31 : Présentation graphique du problème de tournées de véhicules VRP

III.5.4 Différents des modèles de VRP :

La diversité de problème de routage de véhicule vise à étudier des situations plus réalistes de pendant entrez différentes contraintes pour résoudre les problèmes d'une nature quotidienne

a) Le VRP multi-dépôts

Dans ce type Les clients sont desservis par plusieurs dépôts. Chaque dépôt ayant sa propre liste de véhicules. Chaque véhicule part d'un dépôt et retourne au dépôt initial. L'objectif du MDVR(VRP multi-dépôts) est de réduire le nombre de véhicules, la période de voyage et la demande globale de produits qui doivent être distribués à partir de divers dépôts.[22]

b) Le VRP avec fenêtres temporelles

Dans VRP avec fenêtres temporelles, pour chaque produit de consommation doit être distribué dans une certaine fenêtre de temps avec des demandes connues avec minimal coût et la distance. Les véhicules ne peuvent arriver plus tôt ou plus tard que le temps.[22]

c) Le VRP avec livraison/ramassage

les marchandises sont distribuées du centre de distribution à un groupe de clients, et collection résidus de matériaux leur fourni aux centres de distribution dans les mêmes véhicules avec une capacité limitée et d'une manière optimal, C'est à dire que le flux de marchandises doit être aux clients de aller accompagné par la collection de résidus de retour.[22]

d) Le VRP avec contraintes de capacité

Le VRP avec contraintes de capacité (CVRP), est une flotte de véhicules de capacité finie, basée dans un dépôt, doit assurer des tournées entre plusieurs clients (ou villes) ayant demandé chacun une certaine quantité de marchandises. L'ensemble des clients visités par un véhicule désigne la tournée de celui-ci. Chaque client doit être desservie une et une seule fois et chaque tournée commence et se termine au dépôt.[20]

III.5.5.Modélisation mathématique du problème

L'objectif du CVRP est de minimiser le coût total, c.-à-d. la somme des distances ou des temps de parcours des tournées, tout en respectant la contrainte de capacité des véhicules : la quantité de marchandises livrées sur une tournée ne doit pas dépasser la capacité du véhicule qui l'assure.[22]

➤ **Le Modélisation informelle :**

Données : Un ensemble de nœuds client (ayant une demande) et d'arêtes et une flotte illimitée de véhicules de capacité uniforme Q partant d'un unique dépôt.

- **La demande :** Une demande déterministe où la quantité de produits à livrer pour chaque client est connue

Trouver : Des tournées de véhicules, satisfaisant chaque demande une et une seule fois et respectant les contraintes de capacité.

Minimisant : Le coût total de transport lié aux arcs empruntés par les véhicules et/ou les coûts fixes associés à l'utilisation des véhicules.

➤ **La fonction objectif**

$$\text{Minimiser } \sum_{k=1}^m \sum_{i=1}^n \sum_{j=1}^n \text{dist}_{ij} * y_{ijk} \quad (1)$$

➤ **Les contraintes de problème**

$$\sum_{i=1}^{i=n} d_i * y_{ik} \leq \text{cap} \quad k = 1 \dots m \quad (2)$$

$$\sum_{k=1}^m y_{ik} = 1 \quad i = 1 \dots n \quad (3)$$

$$\sum_{i=1}^n x_{ijk} = y_{kj} \quad j = 1 \dots m, \quad \forall k \quad (4)$$

$$\sum_{j=1}^n x_{ijk} = y_{ki} \quad i = 1 \dots n, \quad \forall k \quad (5)$$

$$x_{ijk} \in 0,1 \quad \forall 0 \leq i, j \leq n; 1 \leq k \leq m \quad [21]$$

➤ **Les Paramètres**

i = l'index du client ($I = 1, 2, \dots, n$).

n = nombre de clients.

d_0 = dépôt .

k = l'index du véhicule ($k = 1, 2, \dots, m$).

m = est le nombre de véhicules.

d_i = la demande du client i .

$dist_{ij}$ = distance entre la ville i et la ville j .

c = la capacité de véhicule.

y_{ik} = variable binaire : sa valeur est 1 si le client i est livré en le véhicule k ; autrement il est 0

x_{ijk} = variable binaire qui prend la valeur 1 si la ville i est visité immédiatement avant la ville j . Sinon, cette variable prend la valeur 0.

là où l'équation (1) indique que l'objectif la fonction du problème de VRP est de réduire au minimum toute la distance, l'équation de contrainte (2) commandes que la charge de chaque véhicule pas excéder sa capacité ; l'équation de contrainte (3) signifie que chaque demande de client doit être satisfait en un véhicule, les équations de contrainte (4) et (5) s'assurent qu'il y a seulement un véhicule ce qui arrive et part d'un client.[21]

III.5.5 application de l’algorithme BH pour la résolution du problème de tourné de véhicules

Cette partie est consacrée à la présentation des résultats de l’application de BHA pour la résolution du problème de routage ou tourné de véhicules. Aussi une étude comparative avec d’autres algorithmes (SA et PSO) et menée pour qualifier cet algorithme.

On commence avec une déclaration des paramètres de réglage

a) Déclaration des paramètres :

Les paramètres sont ajustés comme suit :

Tableau III.15 : les paramètres de modèles 1

	Nombres des clients								Nombres de véhicules			
	1	2	3	4	5	6	7	8	Capacité de véhicule	1	2	3
Position x	64	20	57	116	27	156	87	94		54	46	49
Position y	55	99	78	40	7	0	74	17				
Distance entre dépôt et le client	23,34	82,41	40,3	31,01	74,40	85,8	23	34,7				
Demandes des clients	21	11	18	20	23	13	21	12				

Tableau III.16: les paramètres de modèles 2

	Nombres des clients										
	1	2	3	4	5	6	7	8	9	10	11
Position x	140	0	55	136	73	195	141	55	75	106	129
Position y	13	18	36	78	25	41	21	51	3	58	91
Distance entre dépôt le client	68,4	93,62	36,35	55	32,7	109,89	64,66	31,25	53	20,2	56
Demandes des clients	15	12	15	21	12	14	17	10	24	16	21

<div style="text-align: center;">  Nombre des clients </div>										
12	13	14	15	16	17	18	19	20	21	22
138	13	169	191	34	55	198	184	139	23	182
79	96	51	77	76	84	78	42	13	48	34
57,27	83,72	83	107,28	56	42,44	114,33	98,85	67,62	63,38	98,27
13	17	10	25	17	11	13	24	14	12	22

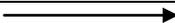
<div style="text-align: center;">  Nombre des clients </div>								Capacité de véhicule	Nombres de véhicules				
23	24	25	26	27	28	29	30		1	2	3	4	5
5	169	199	36	35	143	130	57		116	111	106	109	118
33	10	2	81	39	49	76	44						
83,93	94,4	125	56,35	53,45	57,31	48,75	31						
21	18	22	21	15	18	14	25						

Tableau III.17 : les paramètres de modèle 3

	Nombres des clients										
	1	2	3	4	5	6	7	8	9	10	11
Position x	143	101	123	199	116	15	120	112	28	68	85
Position y	94	22	38	93	37	13	57	74	18	3	38
Distance entre dépôt et le client	62,93	28,16	27,73	109,77	22,2	90,87	23,08	27,78	76,96	55,75	17,69
Demandes des clients	20	18	11	10	25	17	12	18	25	15	15

<div style="text-align: center;">  Nombres des clients </div>												
12	13	14	15	16	17	18	19	20	21	22	23	24
116	56	17	25	79	110	132	62	94	48	196	17	146
80	40	34	29	35	78	31	61	50	70	99	15	95
34,98	43,17	82,56	75,96	24,2	30,46	38,94	37,64	4	53,85	109,56	88,23	65,79
13	21	18	24	18	12	23	14	10	20	12	22	18

<div style="text-align: center;">  Nombre des clients </div>												
25	26	27	28	29	30	31	32	33	34	35	36	37
6	197	64	88	182	72	31	190	192	197	143	121	192
42	34	78	86	19	11	38	50	39	49	7	62	76
92,34	100,3	44,04	37,36	89,53	46,87	68,06	92	94,64	99	62,24	25,94	97,52
18	11	25	14	15	23	10	15	10	16	13	22	25

<div style="text-align: center;">  Nombres des clients </div>												
38	39	40	41	42	43	44	45	46	47	48	49	50
82	3	172	145	183	81	82	3	172	145	183	81	82
27	73	48	2	72	19	27	73	48	2	72	19	27
28,01	97,74	74,02	67,17	87,8	35,35	28,01	97,74	74,02	67,17	87,8	35,35	28,01
20	24	10	23	12	20	20	24	10	23	12	20	20

<div style="text-align: center;">  Nombres des clients </div>												
51	52	53	54	55	56	57	58	59	60	61	62	63
70	199	67	21	18	35	167	169	82	35	113	67	109
63	36	30	59	23	90	54	90	42	86	70	24	16
30,87	101,9	36,89	77,52	84,43	74,62	69,11	81,49	17,88	72,56	25	40,45	35,73
19	19	13	17	25	25	15	17	11	19	24	20	24

Nombres des clients							Nombres de véhicules								
64	65	66	67	68	69	70	Capacité de véhicule	1	2	3	4	5	6	7	8
77	0	91	136	56	108	61		167	160	173	162	172	165	166	189
61	22	66	100	48	32	78									
23,7	101,91	17,46	62,8	42,04	20,59	46,4									
16	21	12	17	23	24	10									

III.6. Résultats de simulation

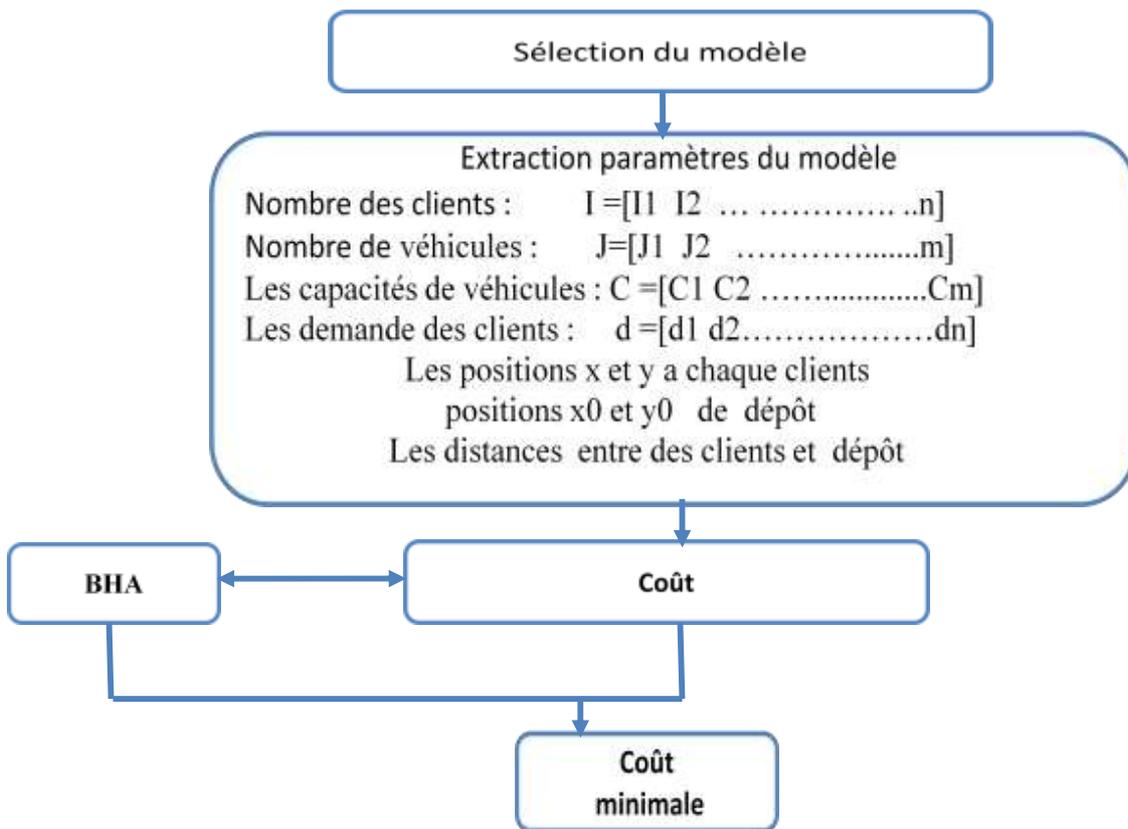


Figure III.32 : Organigramme d’optimisation par BHA

Les tableaux suivants (18,19 et 20) représentent les valeurs des fitness obtenus pour le 1er, le 2ème et le 3ème modèle en fonction d'itération, en utilisant 3 méthodes :

Modèle : est un ensemble de véhicules, de routes et dépôt, et représente en graph

a) 1^{ère} modèle

Tableau III.18 : Meilleurs coûts obtenus du 1^{er} modèle

	PSO	SA	BHA
50 itérations	252.8094	222.8493	22.8493
200 itérations	244.49529	220.1634	220.1634
1000 itérations	227.4293	220.1634	220.1634
3000 itérations	222.8493	220.1634	220.1634

b) 2^{ème} modèle

Tableau III.19 : Meilleurs coûts obtenus du 2^{ème} modèle

	PSO	SA	BHA
50 itérations	983.8375	667.1894	635.8252
200 itérations	990.4557	359.459	589.1918
1000 itérations	975.0975	355.2541	537.0565
3000 itérations	941.725	344.5186	569.7539

c) 3^{ème} modèle

Tableau III.20: Meilleurs coûts obtenus du 3^{ème} modèle

	PSO	SA	BHA
50 itérations	1728.7624	1047.8787	1164.677
200 itérations	1680.8564	547.8601	1010.9159
1000 itérations	1566.5693	387.5515	869.2337
3000 itérations	1367.1513	377.4113	909.04

a) 1^{ème} modèle

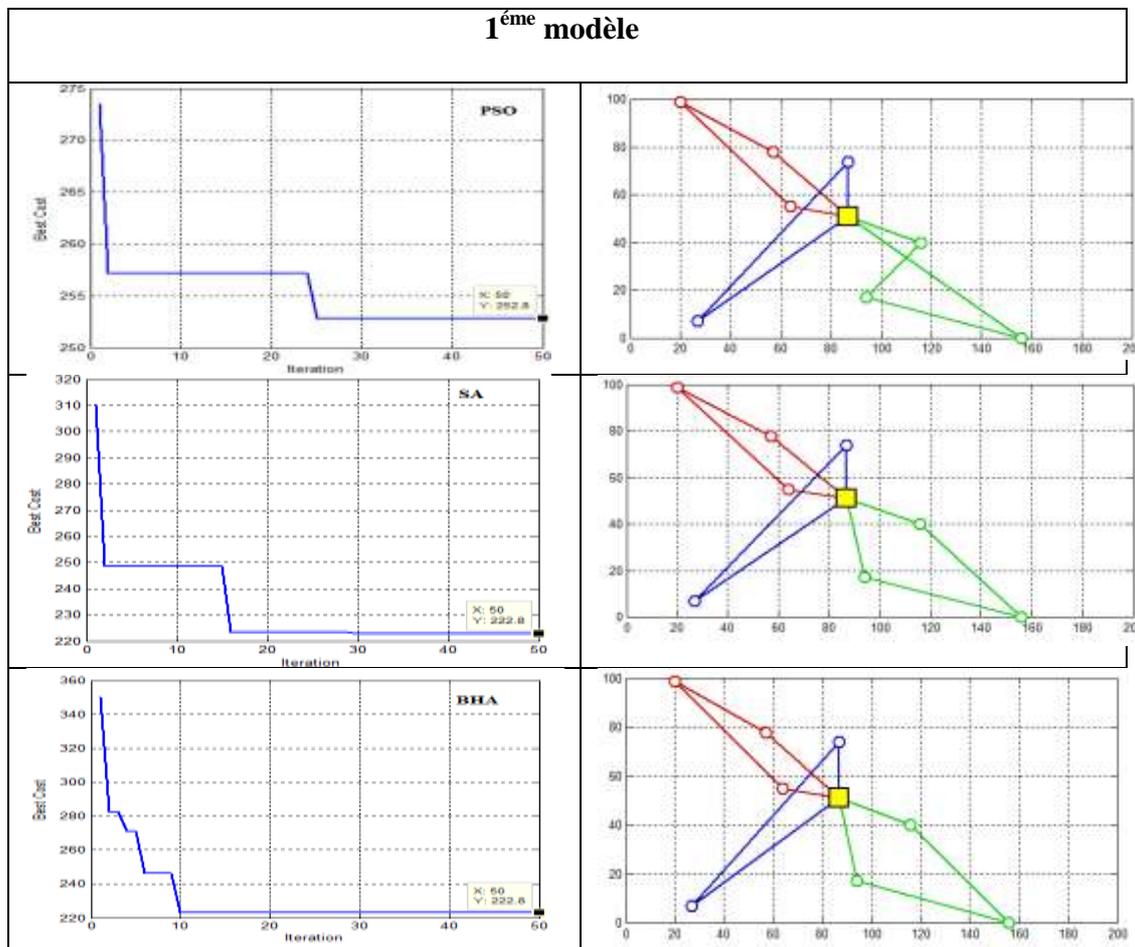


Figure III.33: Résultats de variation de coût en fonction d'itérations (50 itérations) et la forme de modèle 1.

La figure 32 présente la variation de fitness en fonction d'itérations et la forme de modèle. Pour les deux méthodes BHA et SA, nous remarquons la valeur de fitness sont diminuée et converge vers la valeur optimale; Donc, nous remarquons la forme du modèle prend une bonne forme et claire pour les routes après la solution.

Cependant, la méthode PSO est diminuée et converge vers la valeur 252, ainsi, fixées; Donc la forme de ce modèle à pris moins beau forme que les premiers modèles.

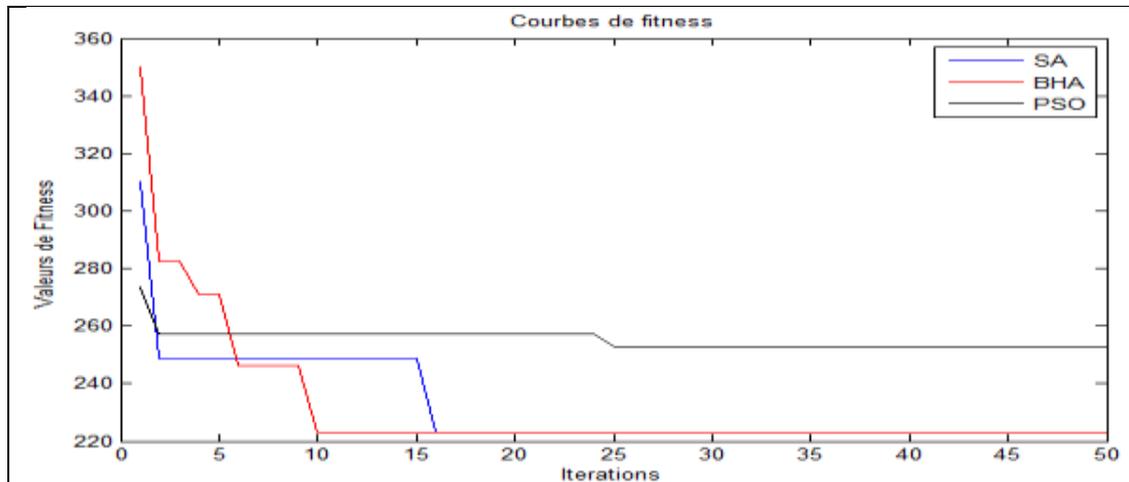


Figure III.34: courbes comparatives des coûts en fonction d'itérations (50 itérations) pour les différentes méthodes (PSO, SA, BHA).

La figure 33 présente les résultats de variations de fitness en fonction d'itérations pour les différentes méthodes (PSO, SA, BHA). Nous remarquons que la valeur de fitness de la méthode BHA est diminuée et converge rapidement vers la valeur optimale à la 10^{ème} itération; mais la méthode SA à la 16^{ème} itération. Cependant, la valeur de fitness de la méthode PSO est loin de la valeur optimale.

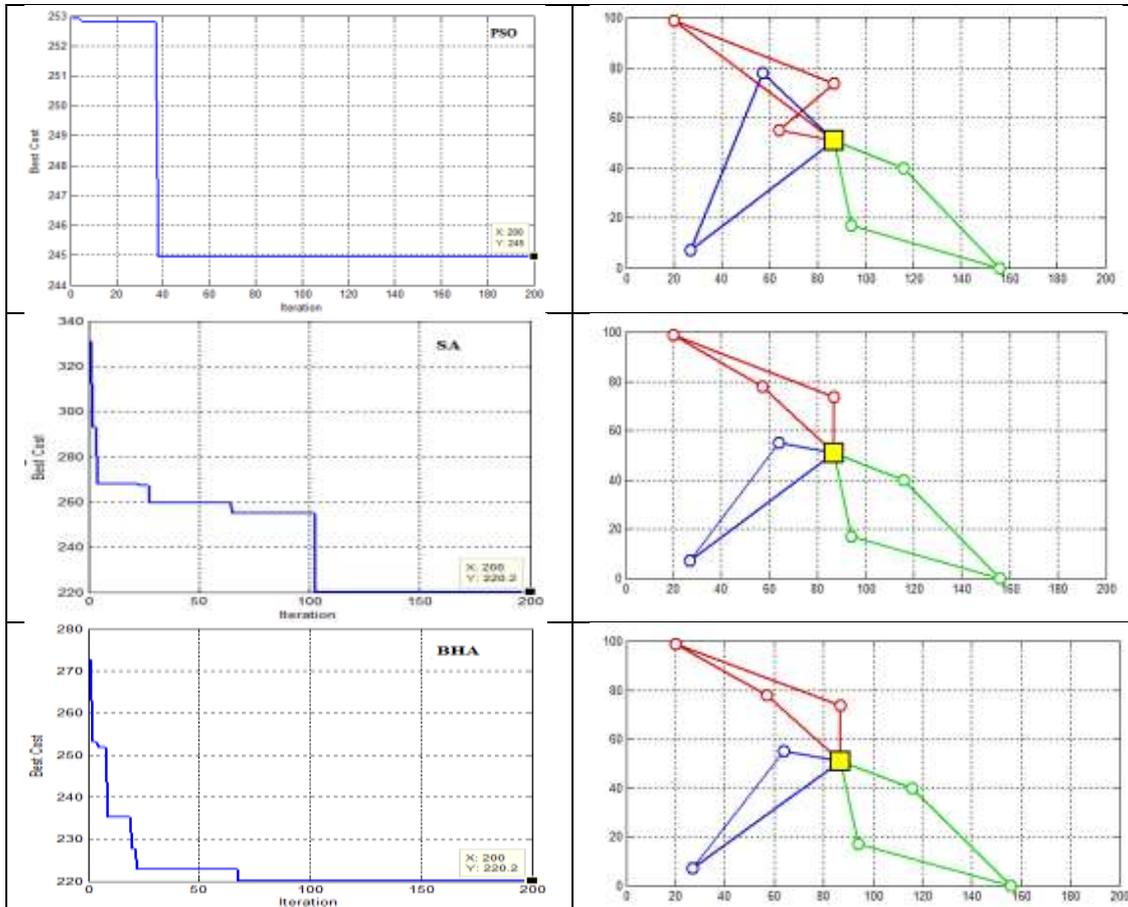


Figure III.35: Résultats de variation de coût en fonction d'itérations (200 itérations).
et la forme de modèle 1.

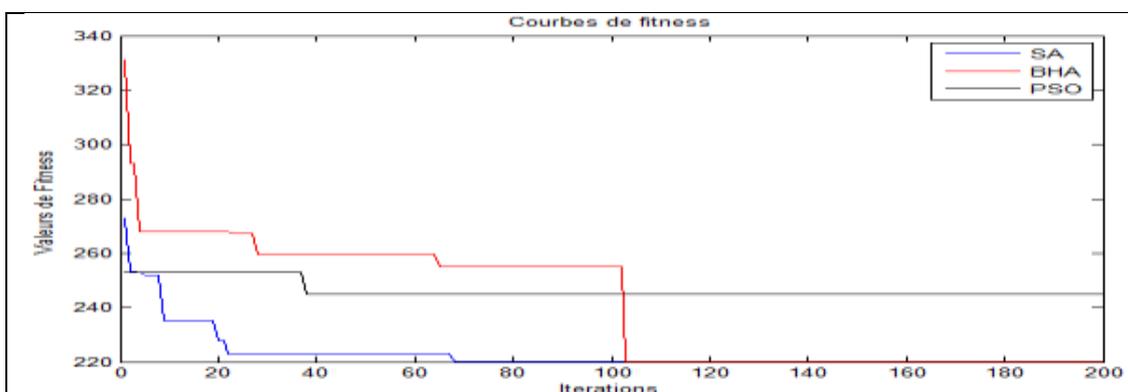


Figure III.36: courbes comparatives des coûts en fonction d'itérations (200) pour les différents méthodes (PSO, SA, BHA).

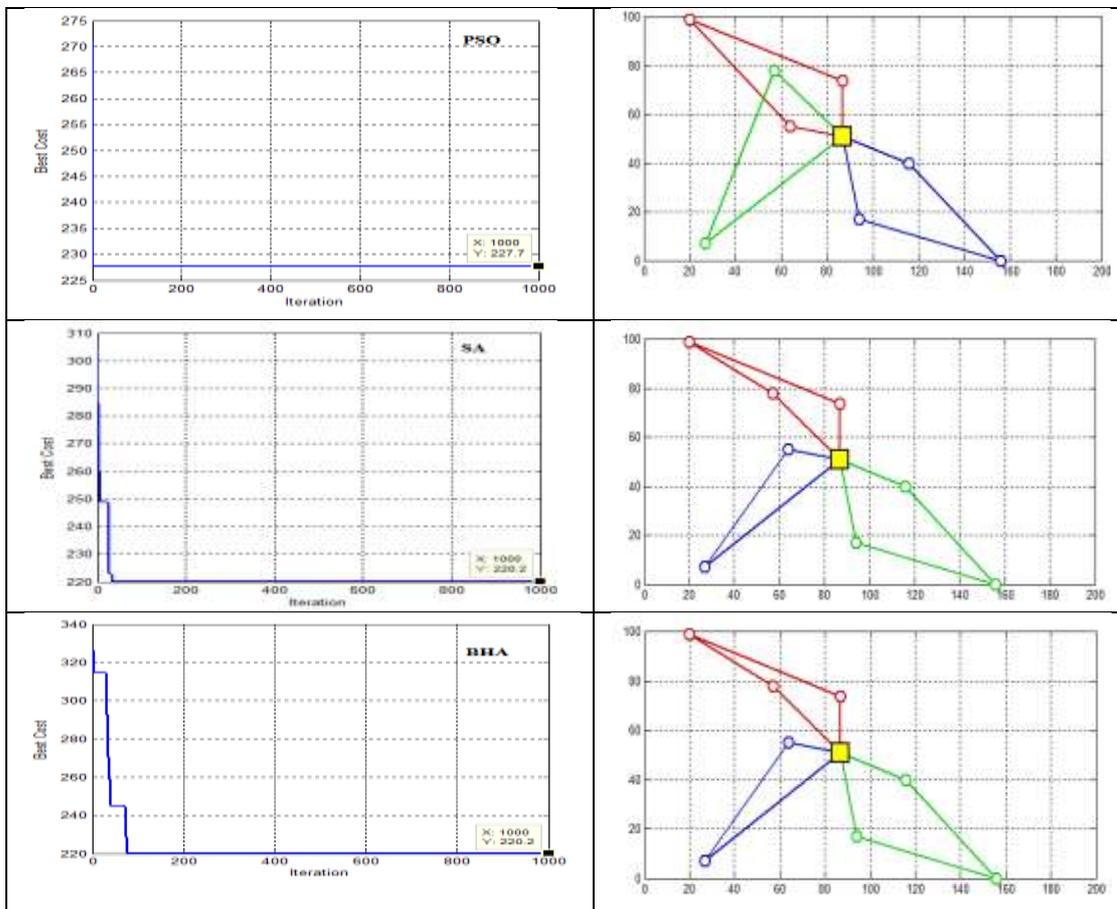


Figure III.37: Résultats de variation de coût en fonction d'itérations (1000 itérations) et la forme de modèle 1.

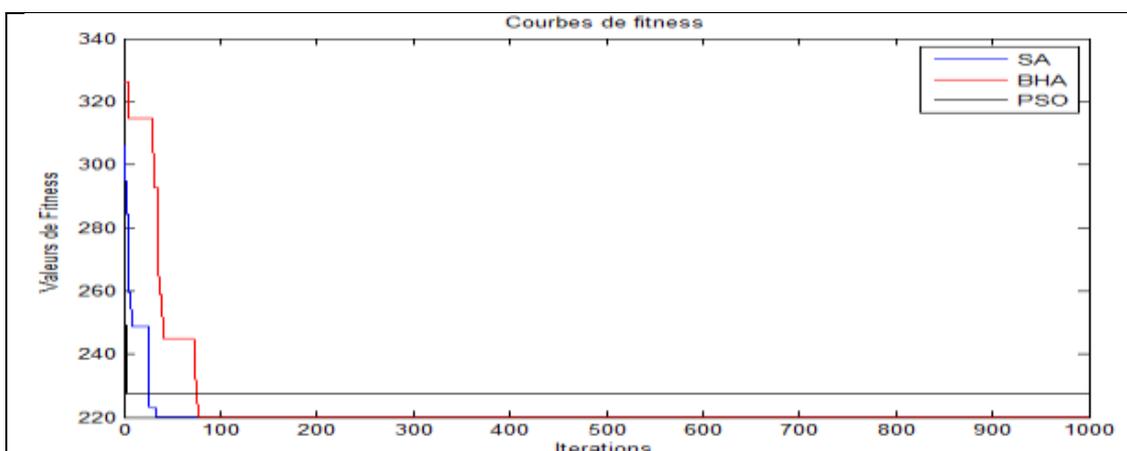


Figure III.38: Résultats de variation de coût en fonction d'itérations (200) pour les différents méthodes (PSO, SA, BHA).

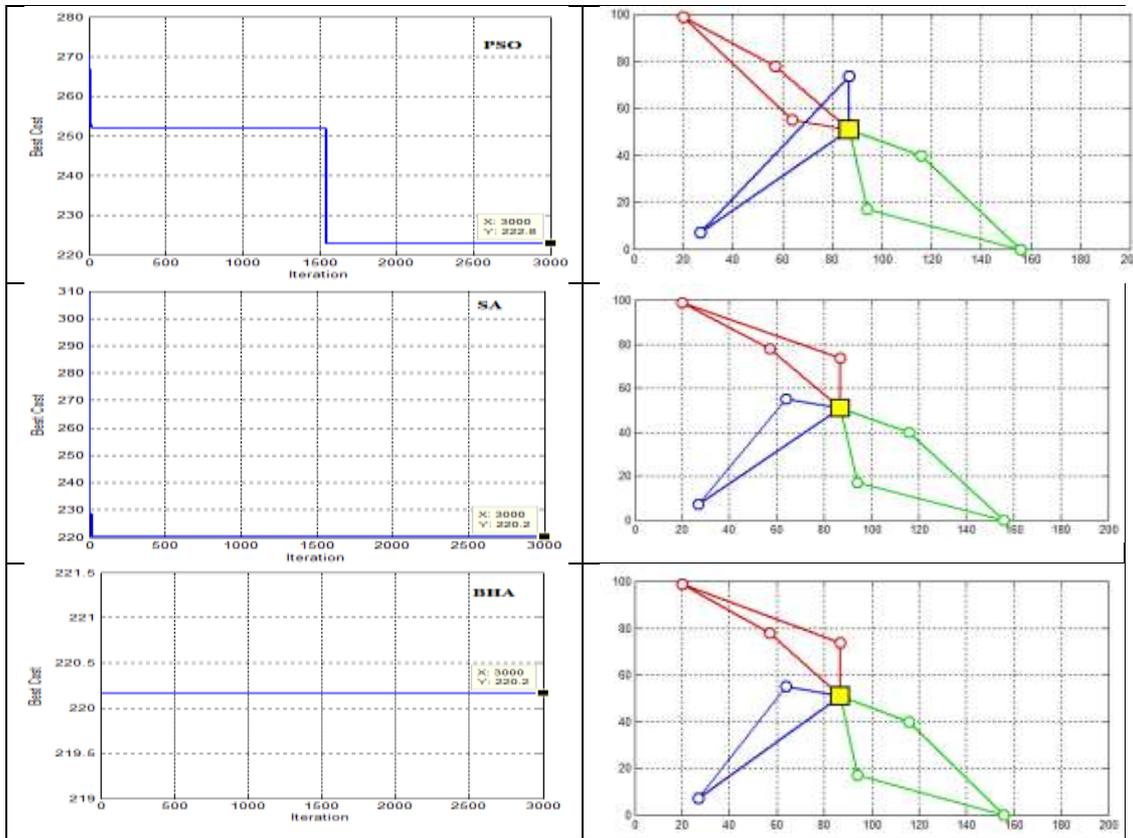


Figure III.39: Résultats de variation de coût en fonction d'itérations (3000 itérations) et la forme de modèle 1.

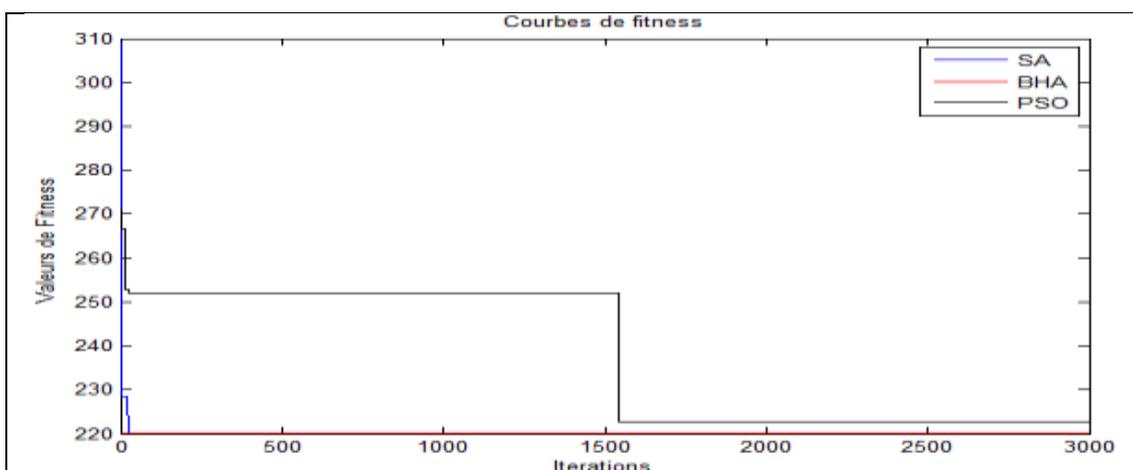


Figure III.40: courbes comparatives des coûts en fonction d'itérations pour les différents pour méthodes (PSO, SA, BHA).

Les figures 34,36 et 38 présentent des variations de coût en fonction d'itérations et la forme de modèle 1; Pour les deux méthodes BHA et SA, nous remarquons que les valeurs des coûts diminuent et convergent vers les valeurs optimales minimales; Donc, nous remarquons que la forme du modèle prend une bonne forme et claire pour les routes après la solution.

Cependant, la méthode PSO diminue et converge vers la valeur loin de la valeur optimale, mais, optimisée à chaque fois qu'on augmente le nombre d'itérations. Donc la forme de ce modèle a pris moins bonne forme que les premiers modèles type 1.

Les figures 35,37 et 39 présentent les résultats de variation de coûts en fonction d'itérations pour les différentes méthodes (PSO, SA, BHA). Nous remarquons que la valeur de coût des méthodes BHA et SA diminuent et convergent rapidement vers la valeur optimale.

Cependant, la valeur de coût de la méthode PSO est toujours loin de la valeur optimale.

b) 2^{ème} modèle

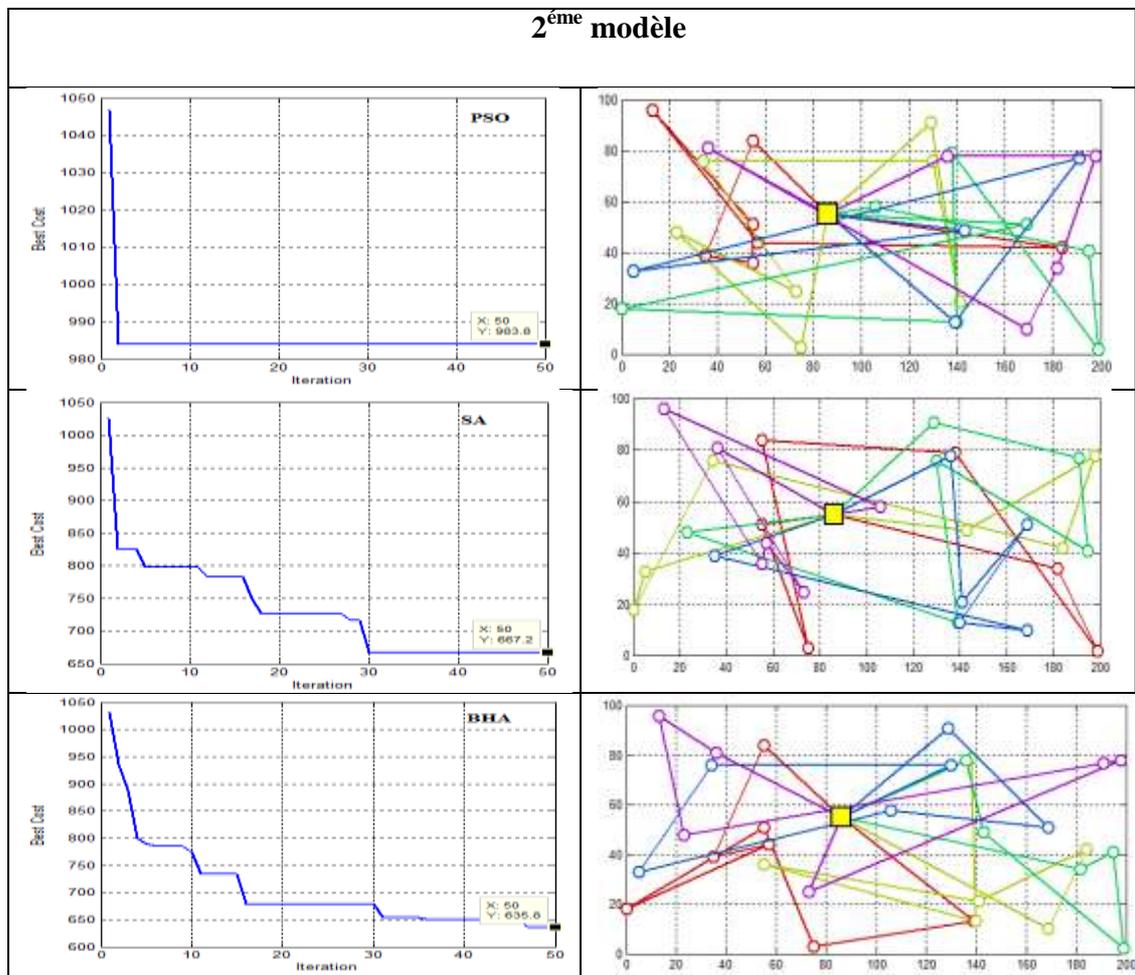


Figure III.41: Résultats de variation de coût en fonction d'itérations (50 itérations).
et la forme de modèle 2.

La figure 40 présentent variation de fitness en fonction d'itérations et la forme de modèle 2. Nous remarquons que la valeur de fitness est diminuée et converge vers la valeur 630 à la 32^{ème} itération pour la méthode BHA et la valeur 667 à la 30^{ème} itération pour le méthode SA; Donc, nous remarquons que la forme du modèle prend une forme pas bien pour les routes après la solution; Parce que ce modèle est de complexité moyenne (Augmentation du nombre de véhicules et de villes).

Cependant, la méthode PSO est diminuée vers la valeur 983 à la 5^{ème} itérations, ainsi, fixées; Donc la forme de ce modèle à pris moins de la forme que les premiers modèles type 2.

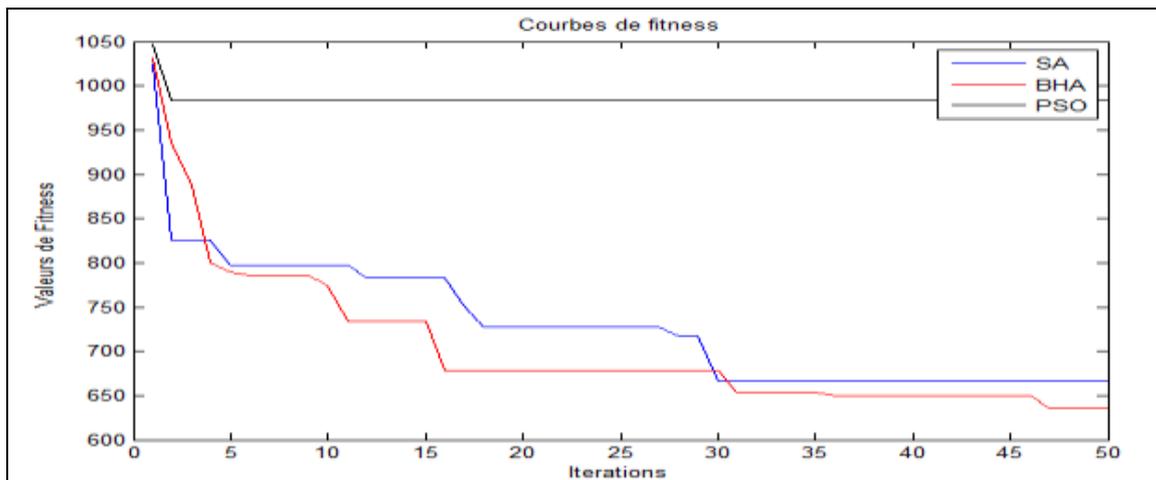


Figure III.42: courbes comparatives des coûts en fonction d'itérations pour les différents pour méthodes (PSO, SA, BHA).

Nous remarquons que les valeurs des coûts des méthodes SA et BHA converges rapidement vers les valeurs 667.2 et 635.5 successivement ; où les valeurs sont loin à la valeur optimale. Mais la valeur de fitness da la méthode PSO est encore plus loin de la valeur de coût des deux méthodes (BHA, SA); Donc la plus loin que la valeur optimale.

Conclusion1 : 50 itérations sont insuffisantes pour obtenir le résultat optimal de ce modèle.

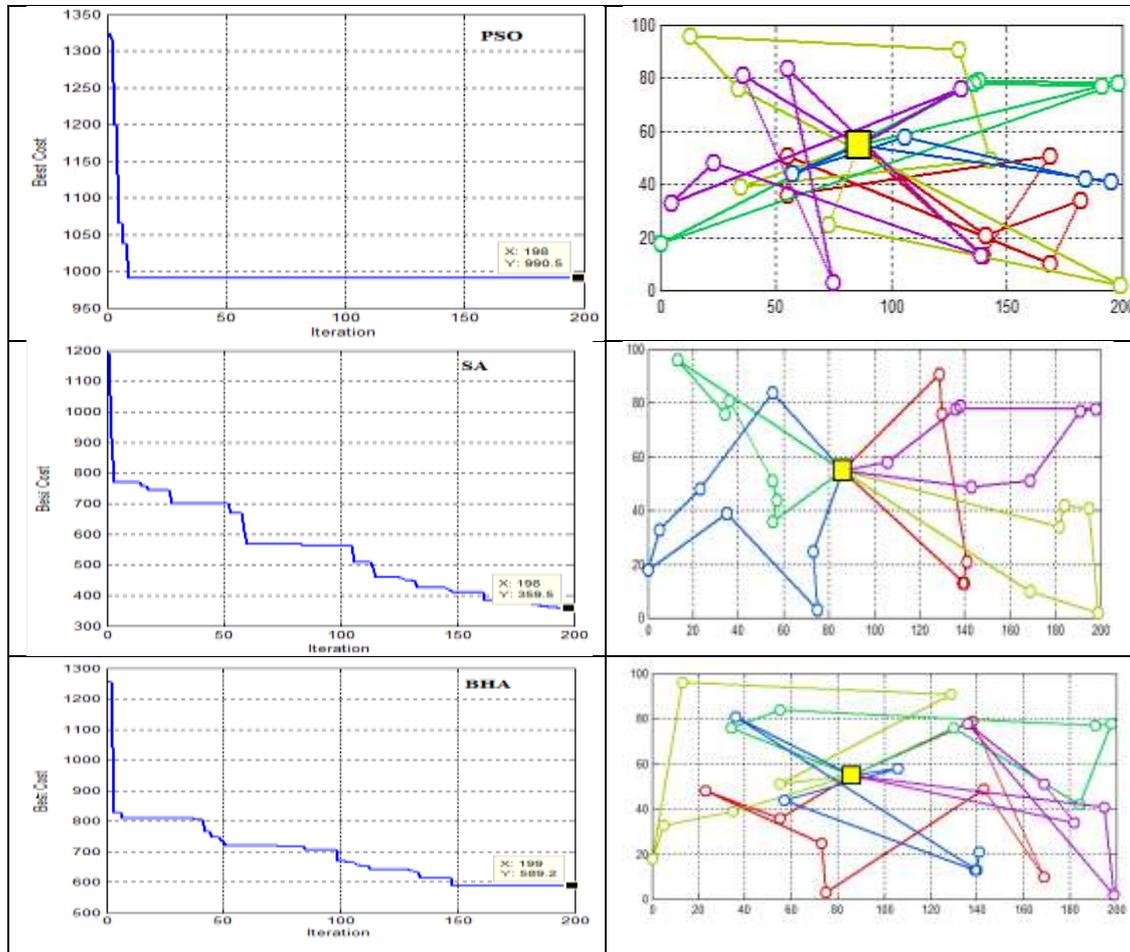


Figure III.43: Résultats de variation de coût en fonction d'itérations (200 itérations).
et la forme de modèle 2.

Nous remarquons que la valeur de coût diminue et converge vers la valeur 589.2 à la 150^{ème} itération pour la méthode BHA et la valeur optimal 359.5 à la 30^{ème} itération pour la méthode SA, donc, nous remarquons la forme du modèle par méthode SA prend une bonne forme et plus claire par rapport à la méthode BHA pour les routes après la solution.

Cependant, la méthode PSO diminue vers la valeur 990.5 à la 10^{ème} itération 5, ainsi, fixées où elle est plus loin de la valeur optimale ; Donc, la forme de ce modèle est moins bonne que celles obtenues par les premiers modèles type 2.

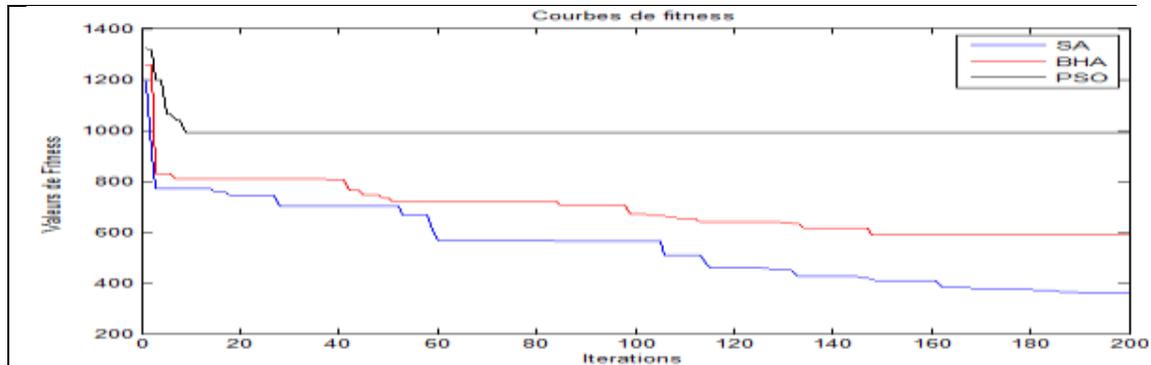


Figure III.44: courbes comparatives des coûts pour les trois méthodes (PSO, SA, BHA).

Nous remarquons que la valeur de coût de la méthode SA converge plus rapidement vers la valeur optimale. Mais la valeur de coût de la méthode BHA converge vers la valeur minimale atteinte qui est égale à 589.2. Elle est la plus proche de la valeur de coût de la méthode SA que le coût obtenu par la méthode PSO

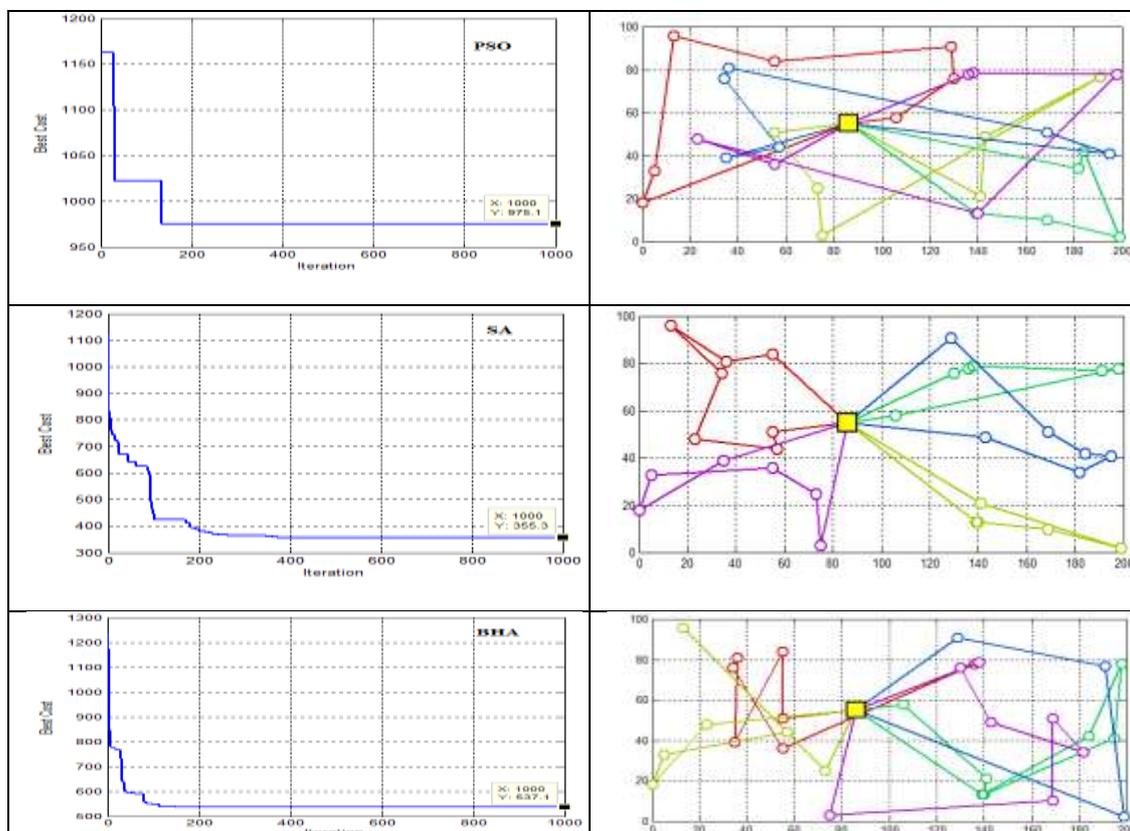


Figure III.45: Résultats de variation de coût en fonction d'itérations (1000 itérations) et la forme de modèle 2.

Nous remarquons que la valeur de coût diminue et converge vers la valeur 537.1 à la 150^{ème} itération pour le méthode BHA et converge vers la valeur optimale 355.5 à la 30^{ème} itération pour le méthode SA; Donc, nous remarquons la forme du modèle par méthode SA prend une bonne forme et plus claire par rapport du méthode BHA pour les routes après La solution.

Cependant, la méthode PSO diminue vers la valeur 975.1 à la 180^{ème} itérations, ainsi, fixées. Elle est plus loin de la valeur optimale. Donc la forme de ce modèle a pris moins de la forme que les premiers modèles.

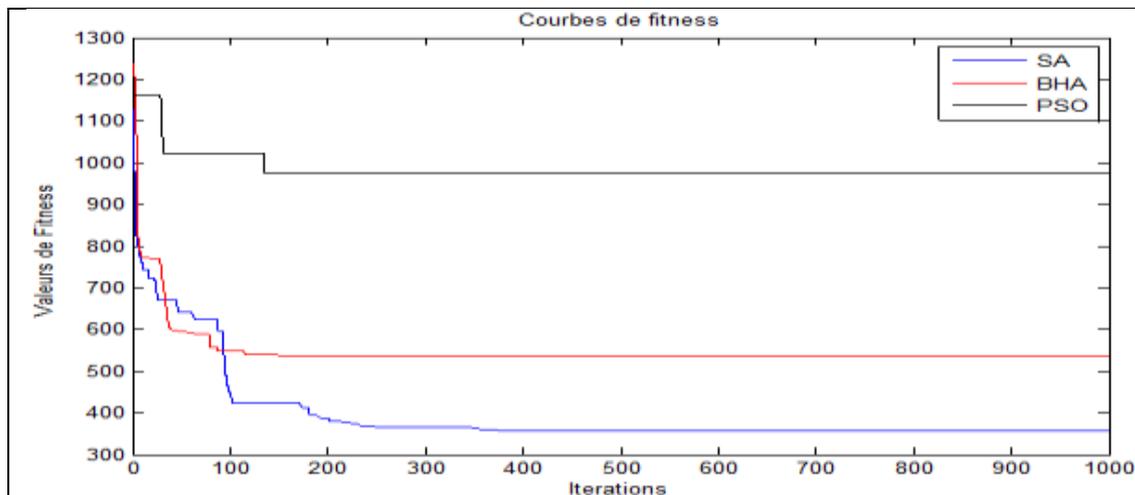


Figure III.46: courbes comparatives des coûts en fonction d'itérations (1000 itérations) pour les différent méthodes (PSO, SA, BHA).

Nous remarquons que la valeur de coût da la méthode SA diminue et converge rapidement vers la valeur optimale et la valeur de fitness de la méthode BHA converge vers la valeur optimale atteinte qui est égale à 537.5. Elle est plus proche de la valeur de coût de la méthode SA que la valeur de coût da la méthode PSO.

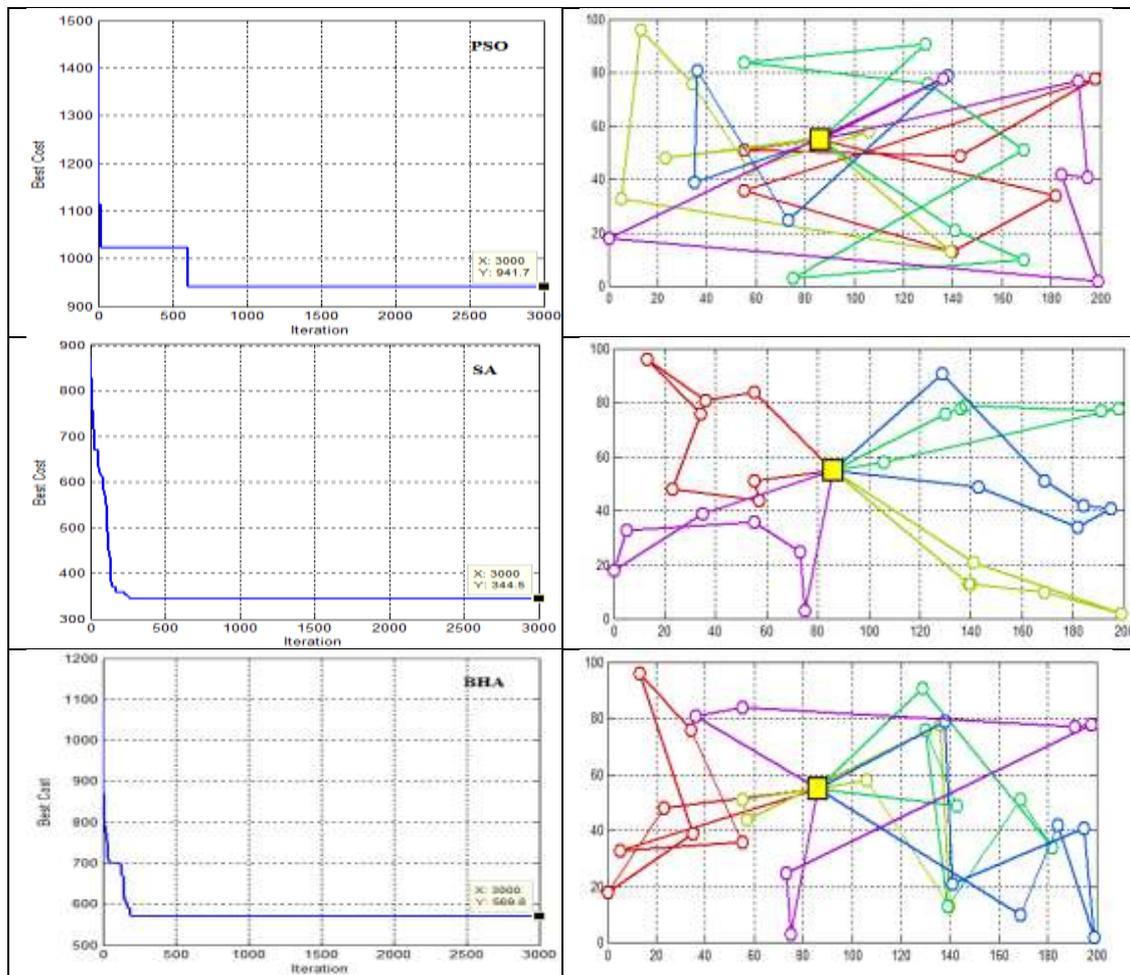


Figure III.47: Résultats de variation de coût en fonction d’itérations (3000 itérations) et la forme de modèle 2.

Nous remarquons que la valeur de coût diminue et converge vers la valeur 569.5 à la 150^{ème} itération pour le méthode BHA et converge vers la valeur optimale 355.5 à la 30^{ème} itération pour le méthode SA; donc, nous remarquons que la forme du modèle par méthode SA prend une bonne forme et plus claire par rapport du méthode BHA pour les routes après la solution.

Cependant, la méthode PSO diminue vers la valeur 941.5 à la 180 itérations, ainsi, fixées. Elle est plus loin de la valeur optimale .Donc la forme de ce modèle a pris moins de la forme que les premiers modèles.

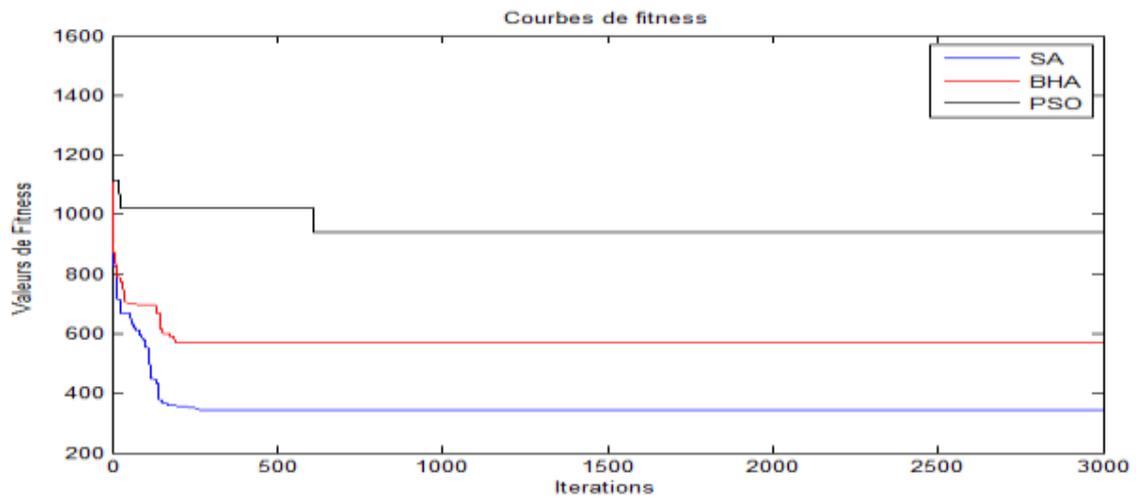


Figure III.48: courbes comparatives des coûts en fonction d'itérations pour les différentes méthodes (PSO, SA, BHA).

Nous remarquons que la valeur de coût de la méthode SA diminue et converge rapidement vers la valeur optimale et la valeur de fitness de la méthode BHA converge vers la valeur minimale atteinte qui est égale à 569.5. Elle est la plus proche de la valeur de coût de la méthode SA que la valeur de coût de la méthode PSO.

c) 3^{ème} modèle

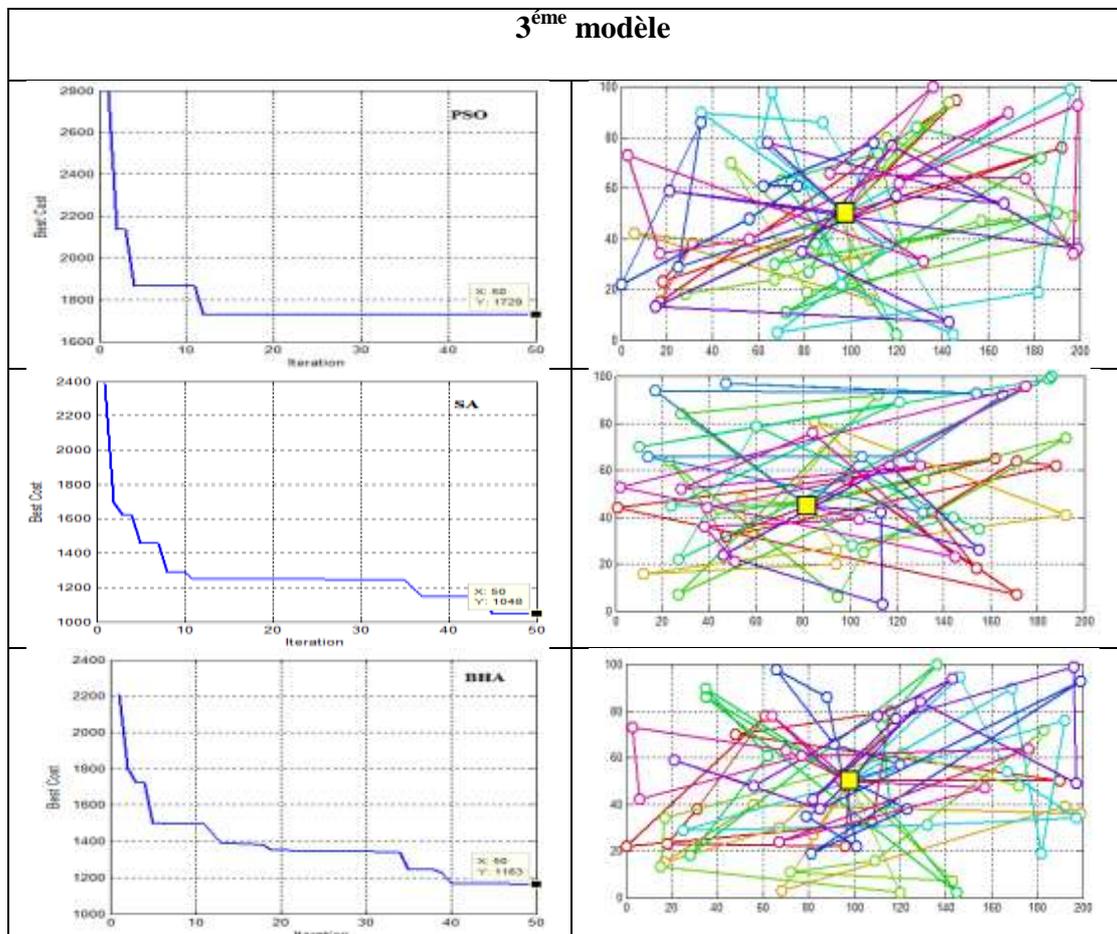


Figure III.49: Résultats de variation de coût en fonction d'itérations (50 itérations).
et la forme de modèle 3.

Nous remarquons que la valeur de coût diminue et converge vers la valeur 1163 à la 40^{ème} itération pour le méthode BHA et la valeur 1048 à la 38^{ème} itération pour la méthode SA; Donc, nous remarquons que la forme du modèle prend une forme pas bien pour les routes après la solution; Parce que ce modèle est de complexité élevée (Augmentation plus du nombre de véhicules et de villes)

Cependant, la méthode PSO diminue vers la valeur 983 à la 5^{ème} itérations, ainsi, fixées; Donc la forme de ce modèle est mauvaise.

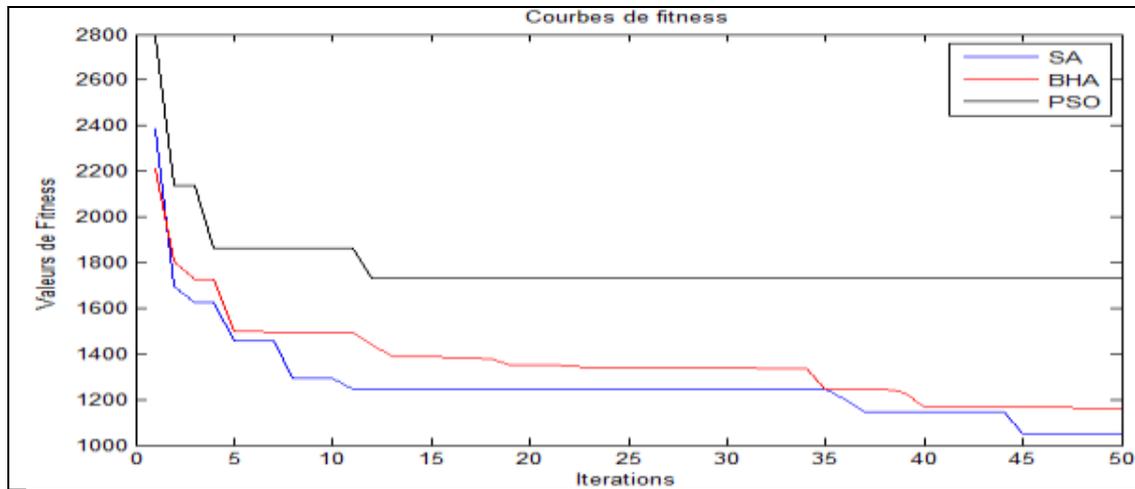


Figure III.50: courbes comparatives des coûts en fonction d'itérations pour les différents méthodes (PSO, SA, BHA).

Nous remarquons que les valeurs de coûts des méthodes SA et BHA convergent rapidement vers les valeurs 1048 et 1163 successivement où les valeurs sont loin à la valeur optimale, mais la valeur de coût de la méthode PSO est encore plus loin des valeurs des coûts des deux méthodes (BHA, SA); Donc, elle est la plus loin de la valeur optimale.

Conclusion2 : 50 itérations sont insuffisantes pour obtenir le résultat optimal de ce modèle.

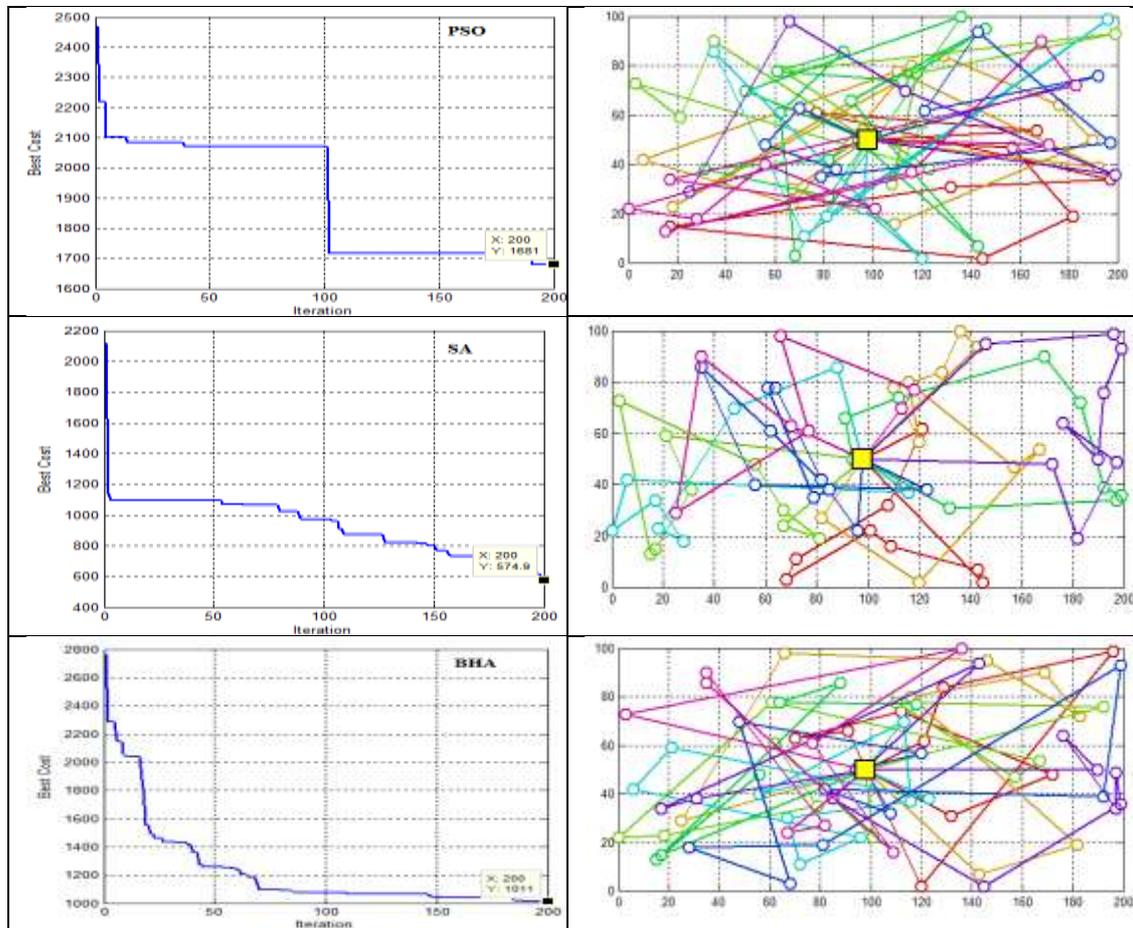


Figure III.51: Résultats de variation de coût en fonction d'itérations (200 itérations).
et la forme de modèle 3.

Dans ce cas, le nombre des itérations sont augmenté pour donner plus de temps aux algorithmes pour réagir. Nous remarquons que la valeur de coût diminue et converge vers la valeur 1011 à la 150^{ème} itération pour la méthode BHA et converge vers la valeur 574.9 à la 160^{ème} itération pour la méthode SA; Donc, nous remarquons que la forme du modèle par méthode SA prend une bonne forme et plus claire par rapport à celui du méthode BHA pour les routes après la solution.

Cependant, la méthode PSO diminue vers la valeur 975.1 à la 180^{ème} itération, ainsi, fixées, elle est toujours plus loin de la valeur optimale ; Donc la forme de ce modèle est plus mauvaise que les formes des premiers modèles.

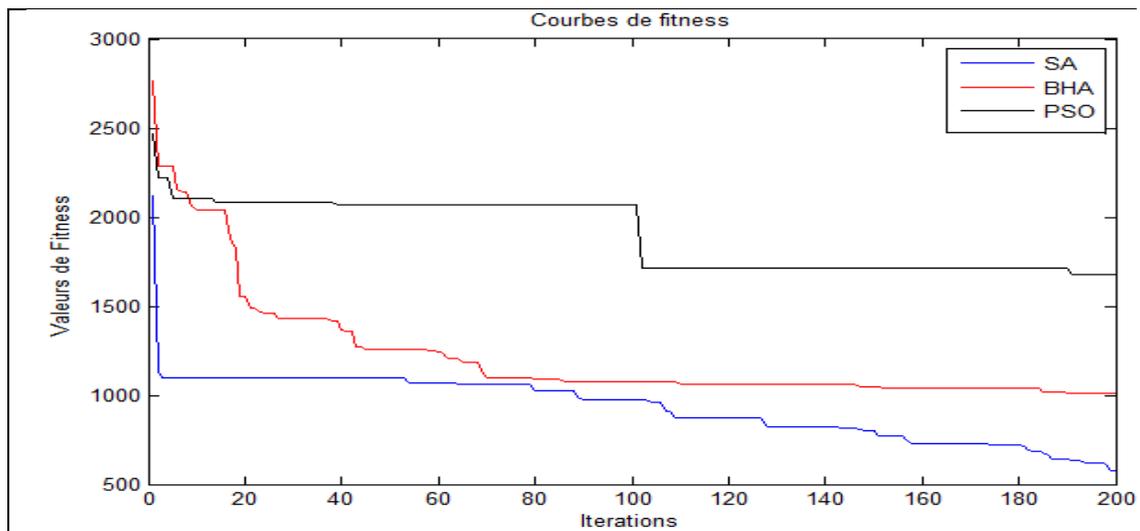


Figure III.52: courbes comparatives des coûts en fonction d'itérations pour les différents méthodes (PSO, SA, BHA).

Nous remarquons que les valeurs de coût des méthodes SA et BHA convergent rapidement vers les valeurs 574.9 et 1011 successivement où les valeurs sont loin de la valeur optimale, mais la valeur de fitness de la méthode PSO est encore plus loin des valeurs de coûts des deux méthodes (BHA, SA); Donc est la plus loin de la valeur optimale.

Conclusion3 : 200 itérations sont insuffisantes pour obtenir le résultat optimal de ce modèle.

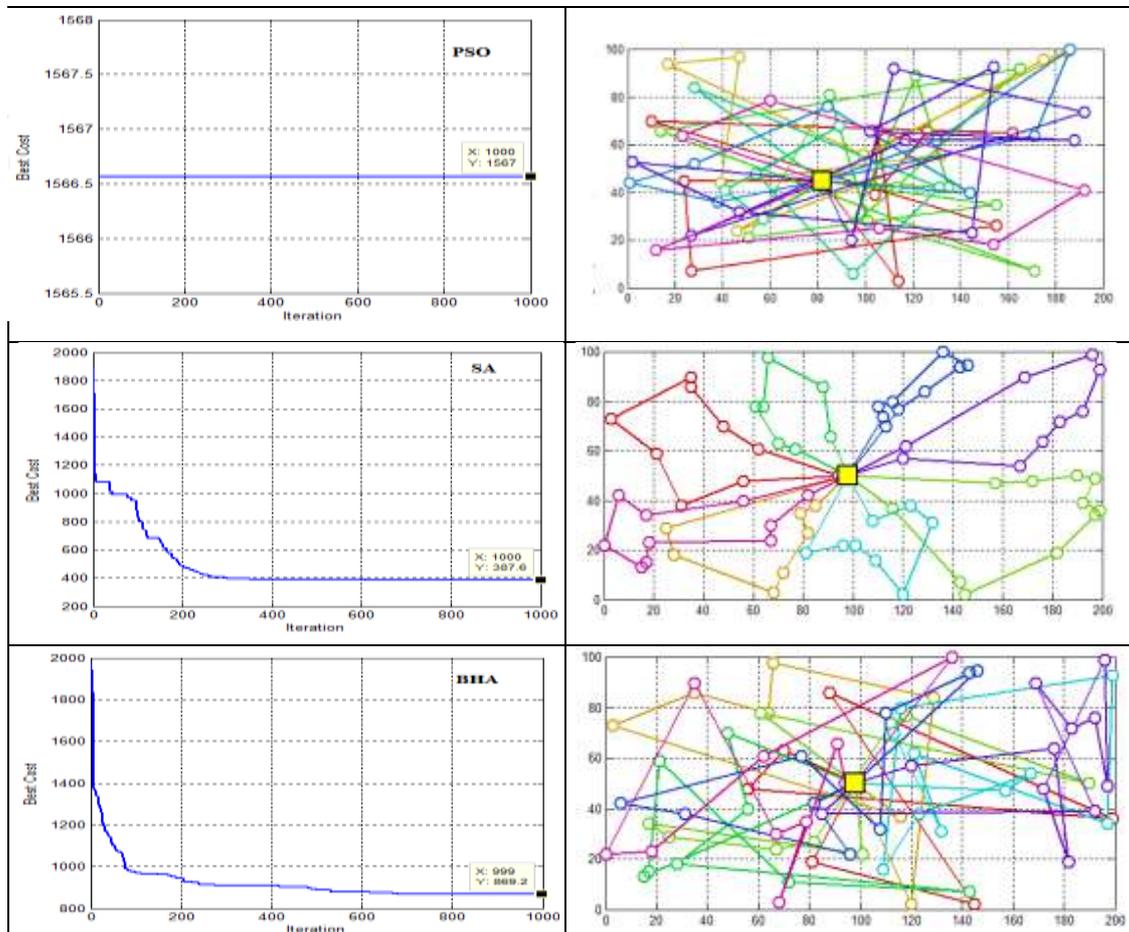


Figure III.53 : Résultats de variation de coût en fonction d'itérations (1000 itérations) et la forme de modèle 3.

Une autre augmentation au nombre des itérations est effectuée. Nous remarquons que la valeur de coût diminue et converge vers la valeur 869.2 à la 550^{ème} itération pour la méthode BHA et converge vers la valeur optimale 387.5 à la 250^{ème} itération pour la méthode SA; donc, nous remarquons la forme du modèle par méthode SA prend une bonne forme et plus claire par rapport du méthode BHA pour les routes après la solution.

Cependant, la méthode PSO est fixée dans la valeur 1567; Donc la forme de ce modèle est mauvaise.

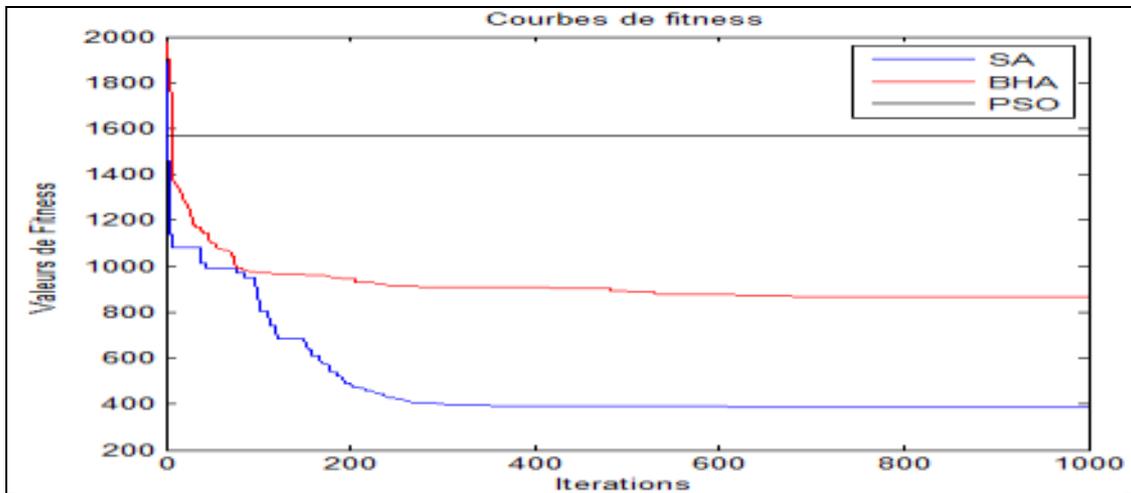


Figure III.54: courbes comparatives des coûts en fonction d'itérations pour les différentes méthodes (PSO, SA, BHA).

Nous remarquons que la valeur de coût de la méthode SA converge rapidement vers la valeur optimale et la valeur de fitness de la méthode BHA converge vers la valeur optimale atteinte qui est égale à 869.2. Elle est la plus proche que la valeur de coût de la méthode SA que la valeur de coût de la méthode PSO

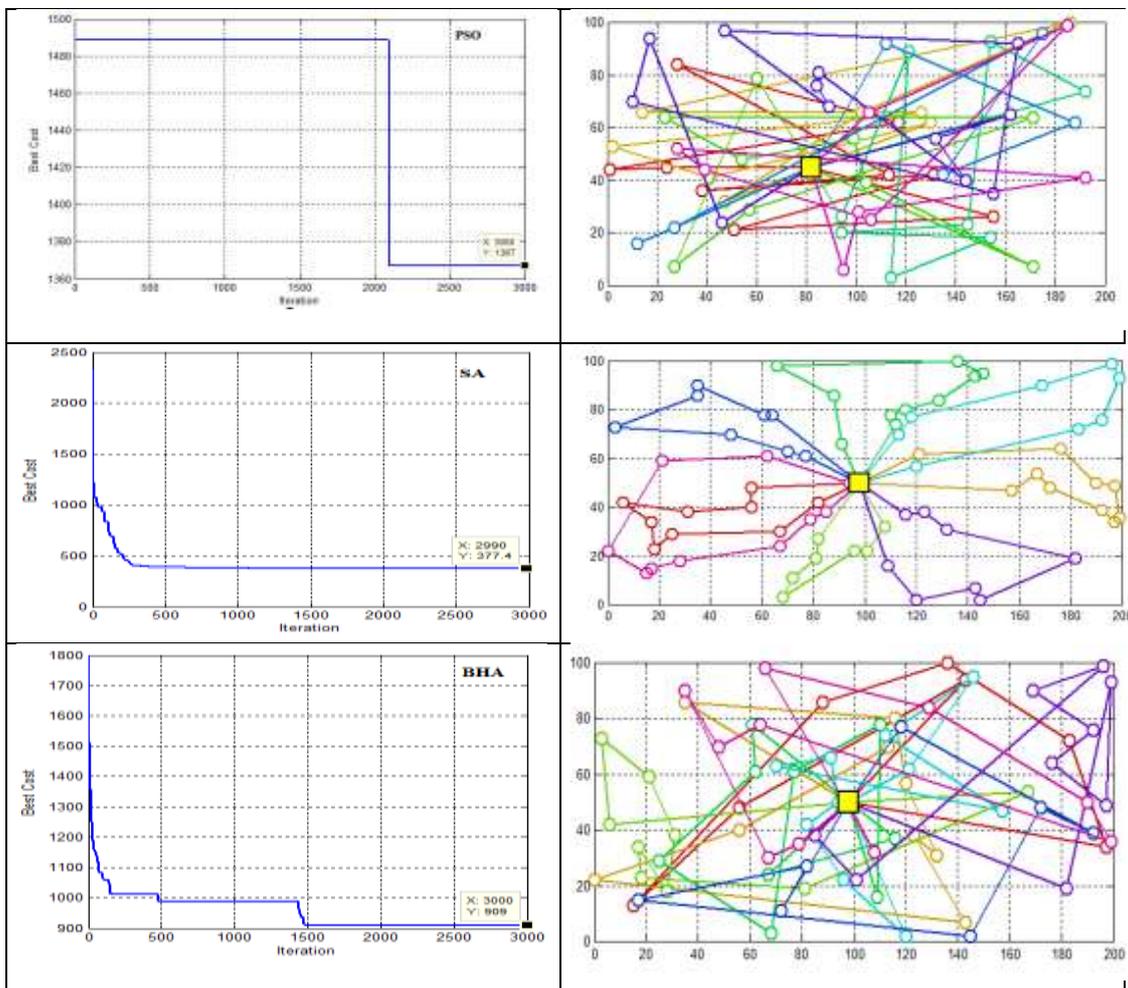


Figure III.55: Résultats de variation de coût en fonction d'itérations (3000 itérations) et la forme de modèle 3.

Pour 3000 itérations, nous remarquons que la valeur de coût diminue et converge vers la valeur 909 à la 1500^{ème} itération pour le méthode BHA et diminue et converge rapidement vers la valeur optimale 377.4 à la 600^{ème} itération pour le méthode SA; Donc, nous remarquons que la forme du modèle obtenue par la méthode SA est bonne et plus claire par rapport à celle obtenue par la méthode BHA pour les routes après la solution.

Cependant, la méthode PSO diminue vers la valeur 1367 à la 2100 itérations, ainsi, fixées; Donc la forme de ce modèle est moins bonne que les formes des autres modèles type 3.

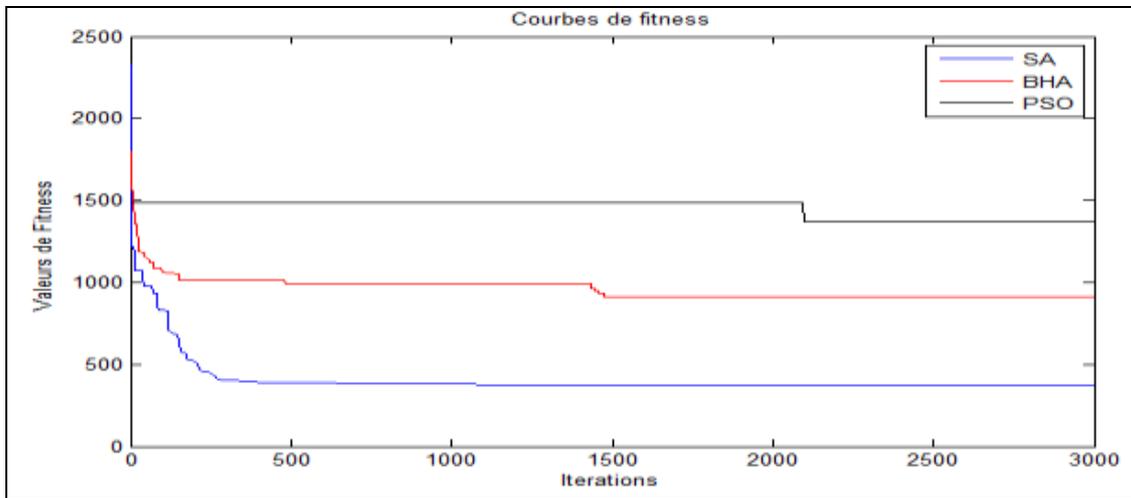


Figure III.56: courbes comparatives des coûts en fonction d'itérations pour les différents méthodes (PSO, SA, BHA).

Nous remarquons que la valeur de coût de la méthode SA converge rapidement vers la valeur optimale et la valeur de coût de la méthode BHA converge vers la valeur optimale atteinte qui est égale à 909. Elle est la plus proche de la valeur de coût de la méthode SA que la valeur de coût de la méthode PSO.

III.7. Conclusion

Dans ce chapitre nous avons appliqué l'algorithme BHA sur deux applications, dont l'objectif est le test de ses performances.

La 1ère a été appliquée sur quelques fonctions de norme célèbres, pour montrer ses performances sur les fonctions de test. Cette application a été accompagnée par une étude de l'effet de changement des paramètres d'initialisation sur la convergence du système et la précision des résultats obtenus ; ensuite appliquer les algorithmes SA et PSO et notre algorithme sur ces fonctions et comparaison entre eux.

Dans la 2ème application nous avons choisi un problème réel, qui est l'optimisation de problème de routage de type CVRP, dont l'objectif est de minimiser le coût total des tournées en respectant la contrainte de capacité des véhicules

Finalement, les résultats de simulation obtenus montrent que la BHA est arrivée à des résultats acceptables, Dans de tels types de problèmes

Conclusion générale

Dans ce mémoire, nous avons étudié une nouvelle méthode d'optimisation métaheuristique, appelée algorithme de trou noir (BHA); elle a été inspirée de caractéristiques de phénomène du trou noir dans l'espace.

Le BHA est un algorithme à base de population, capable de résoudre des problèmes d'optimisation difficiles; ses caractéristiques permettent l'exploitation de l'espace de recherche pour trouver des solutions optimales. Cette fonctionnalité est importante pour les algorithmes qui dépendent sur les populations.

Dans l'application réelle, Nous avons traité le problème de routage de véhicule, de type CVRP; à chaque fois, nous augmentons le nombre de voitures et le nombre de clients pour rendre le problème plus difficile. Le but est trouver la distance la plus courte parcourue par ces véhicules avec respectant les contraintes comme la capacité du véhicule et la quantité de demandes des clients. La méthode BHA a donné des résultats acceptables, dans certaines cas.

Perspectives

Comme future perspectives, nous proposons ;

- ✚ D'appliquer l'algorithme BHA sur d'autres problèmes d'optimisation de différents types (par exemple le traitement d'image).

- ✚ L'Hybridation du BHA avec d'autres algorithmes pour améliorer son fonctionnement.

Bibliographie

- [1] : Benkaddour Halima aribi ramzi, **métaheuristiques parallèles pour la résolution des problèmes difficiles**, mémoire master académique, université kasdi merbah ouargla, 2012/2013.
- [2] : Abbas EL DOR, « Perfectionnement des algorithmes d'optimisation par Essaim Particulaire. Application en segmentation d'image et en électronique », thèse doctorat université PARIS-EST, 5 décembre 2012.
- [3] : Jean-Philippe Préaux, Optimisation Continue,[en ligne]. Disponible sur :
: <https://old.i2m.univ-amu.fr/~preaux/PDF/.../OptimisationContinue/optimisationEA1.pdf>
(Consulté le 20/04/2018).
- [4] : BECHIR, AZIZA. Résolution des problèmes d'optimisation par les systèmes multi-agents et les approches evolutionnaires.Thèse de doctorat. Université Mohamed Khider-Biskra. 2016
- [5] : _Aouag Mouna .Mr.Aalyeb.Youcef Nesrine Chekroud Houda, Les méthodes Hybrides ,Informatique Académique Master 2,[en ligne]. Disponible sur :
: <http://slideplayer.fr/slide/2840207/> (Consulté le : avril 2018).
- [6] : Hacene halim, État de l'art sur l'optimisation combinatoire , Mémoire de Master 2, Université Mohamed Khider-Biskra. [en ligne] Disponible sur :
<http://dspace.univbiskra.dz:8080/jspui/bitstream/123456789/5235/2/m%C3%A9moire.pdf>
(Consulté le : 17/05/2018).
- [7] : Si Tayeb Abdelkader, Méta heuristiques pour l'optimisation des puissances actives dans un réseau d'énergie électrique, Mémoire de Magister, Université de Sciences et Technologie d'Oran, 2010-2011 at, l'université des sciences et de la technologie houari Boumediene, 2013
- [8] : Alaoui Abdiya, Application des techniques des métaheuristiques pour l'optimisation de la tâche de la classification de la fouille de données, mémoire, université des sciences et de la technologie d'oran mohamed boudiaf, 2011/2012.
- [9] : Gherboudj, Amira. "Méthodes de résolution de problèmes difficiles académiques." thèse doctorat .Université de Constantine 2 (2013).

- [10] : Mohamed Zennir, Recherche intelligente et collaborativen ,cours,[en ligne].
Disponible sur : <https://www.cloudschool.org/zennir/recherche-intelligente-et-collaborative>. (Consulté le : 20/05/2018).
- [11] : ilhem boussaïd, perfectionnement de métaheuristiques pour l'optimisation continue, thèse de doctorat, l'université des sciences et de la technologie houari Boumediene, 2013
- [12]: Douiri, M., Souad Elbernoussi, and Halima Lakhbab. "Cours des méthodes de résolution exactes heuristiques et métaheuristiques." Université Mohamed V, Faculté des sciences de Rabat .2009
- [13] : Dr. LEMOUARI Ali, Introduction Aux Métaheuristiques , Support de Cours Introduction Aux Métaheuristiques, Université de Jijel .2014
- [14] HaJJI, Omessaad. Contribution au développement de méthodes d'optimisation stochastiques. Application à la conception des dispositifs électrotechniques. Mémoire de thèse de Doctorat, Université des sciences et technologies de Lille.2003
- [15] : sébastien Noël, métaheuristiques hybrides pour la résolution du problème d'ordonnancement de voitures dans une chaîne d'assemblage automobile, mémoire présenté comme exigence partielle de la maîtrise en informatique, université du Québec à Chicoutimi, octobre 2007.
- [16] : Layeb, Abdesslem. Utilisation des Approches d'Optimisation Combinatoire pour la Vérification des Applications Temps Réel.Thèse de doctorat. université Mentouri de Constantine, 2010.
- [17] : KUMAR, Santosh, DATTA, Deepanwita, et SINGH, Sanjay Kumar. Black hole algorithm and its applications. In : Computational intelligence applications in modeling and control.2015
- [18] : Laurent Sacco (Journaliste).Sciences trou noir [en ligne] Disponible sur : <https://www.futura-sciences.com/sciences/definitions/univers-trou-noir-62/> (Consulté le : mai 2018).
- [19] : HATAMLOU, Abdolreza. Black hole: A new heuristic optimization approach for data clustering. Information sciences 222, 2013 175-184
- [20] : Ben Ismail, S., F. Legras, and G. Coppin. Synthèse du problème de routage de véhicules. Collection des rapports de recherche de Télécom Bretagne .2011.

[21] : LAYEB, Abdesslem, AMMI, Meryem, et CHIKHI, Salim. A GRASP algorithm based on new randomized heuristique for vehicle routing problem. Journal of computing and information technology, 2013.

[22] Yesodha, R., and T. Amudha. A study on bio-inspired metaheuristics for solving vehicle routing problem. Indian Journal of Science and Technology 8.25 .2015