République Algérienne Démocratique Et Populaire

Ministère De L'enseignement Supérieur et de la Recherche Scientifique

Université Saad DAHLAB BLIDA-1

Faculté De Technologie

Département Des Energies Renouvelable

Option : Conversion Photovoltaïque

Mémoire de Fin d'étude pour l'obtention du Diplôme Master 2

Etude comparative entre les algorithmes Méta-heuristique

Présenté par : **KARAALI Chaima**

SELLAM Zineb

Soutenu Devant Le Jury Composé Par :

Monsieur	**DOUMAZ**	**Toufik**	**Président**	**MCB**	**USDB1**
Docteur	**BOUZAKI**	**Moustafa**	**Examinateur**	**MCB**	**USDB1**
Docteur	**KHODJA**	**Fouad**	**Examinateur**	**MCB**	**USDB1**
Docteur	**AIT SAHED**	**Oussama**	**Encadreur**	**MAA**	**USDB1**
Docteur	**BOUKENOU**	**Rachid**	**Co-Encadreur**	**MCB**	**USDB1**

Juillet 2022

ملخص:

الهدف الرئيسي من هذه الأطروحة هو إجراء دراسة مقارنة مفصلة على العديد من خوارزميات التحسين metaheuristic بما في ذلك GA و PSO و ABC.

تم تقييم الخوارزميات المدروسة باستخدام العديد من الوظائف العددية المعيارية في ظل ظروف عمل مختلفة. كنا مهتمين بقياس جودة التصغير وسرعة التقارب وتعقيد الحساب من خلال تغيير حجم السكان وعدد التكرارات القصوى ومدى تعقيد مشكلة التحسين.

أشارت النتائج التي تم الحصول عليها بوضوح إلى أن خوارزمية GA هي الأكثر كفاءة بشكل عام من بين الخوارزميات الثلاثة فيما يتعلق بجودة التحسين وسرعة التقارب والتعقيد الحسابي.

الكلمات المفتاحية: التحسين غير الخطي، الخوارزميات الكونية، التصغير، سرعة التقارب، العشوائية.

Résume

L'objectif principale de ce mémoire est de faire une étude comparative détaille sur plusieurs algorithmes d'optimisation méta-heuristiques notamment le GA, PSO et ABC.

Les algorithmes considérés ont été évalués en utilisant plusieurs fonctions numériques Benchmark sous différentes conditions de travail. On s'est intéressé a mesuré la qualité de minimisation, la vitesse de convergence, et la charge du calcul et cela en variant la taille de la population, le nombre d'itérations maximales et la complexité du problème d'optimisation.

Les résultats obtenus on clairement indiqué que l'algorithme GA est généralement le plus efficace des trois algorithmes vis à vis sa qualité d'optimisation, sa vitesse de convergence et sa complexité du calcul.

Mots clés : optimisation non-linéaire, algorithmes méta-heuristiques, minimisation, vitesse de convergence, aléatoire.

Abstract

The main objective of this thesis is to make a detailed comparative study on several meta-heuristic optimization algorithms including GA, PSO and ABC.

The considered algorithms were evaluated using several Benchmark numerical functions under different working conditions. We were interested in measuring the quality of minimization, the convergence rate , and the complexity of the calculation by varying the size of the population, the number of maximum iterations and the complexity of the optimization problem.

The obtained results clearly indicated that the GA algorithm is generally the most efficient of the three algorithms with respect to its optimization quality, its convergence speed and its computational complexity.

Keywords: nonlinear optimization, meta-heuristic algorithms, minimization, convergence speed, randomness.

Remerciements :

Nous remercions d'abord et avant tout le bon **Dieu** qui nous avons donné la force et la patience pour mener notre travail à terme.

Nous tenons à remercier infiniment notre encadreur Monsieur **Ait Sahed Oussama** docteur en génie électrique à l'Université de Blida1 et notre Co-encadreur **Boukenoui Rachid** docteur en électronique à l'Université de Blida1 d'avoir accepté de diriger notre travail, ses conseils bienveillant, qui nous a encouragés à fournir plus d'efforts pour être à la hauteur de leur attente.

Nos plus sincères remerciements vont aux membres du jury pour l'honneur qu'ils acceptant de juger ce travail.

Encore et avec une grande fierté et honneur que nous tenons à présenter nos remerciement a tout la famille administrative du département des énergies renouvelables, à tous ces enseignants qui nous ont permis d'acquérir des connaissances.

Dédicace :

J'ai le grand plaisir de dédier ce modeste travail :

A l'homme de ma vie, mon exemple éternel ***'PAPA'*** *j'espère que tu es au paradis.*

*A la lumière de mes jours, la source de mes efforts, ma vie et mon âme, la prunelle de mes yeux, a la femme qui peut être fier et trouver ici le résultat de longue années de sacrifices et de privations pour m'aider à avancer dans la vie ma chère maman "****Aida****". A mes chères sœurs '****Zouleikha****', '****Nousseiba****', '****Salssabil****' et '****Romaissa****' qui m'avez toujours soutenu et encouragé durant ces années d'étude.*

A mes chers frères ' ***Fawzi****', '****Ahmed****' et '****Ali****'.*

*A ma chère binôme "****Sellam Zineb*** *" qui a été pour moi la sœur que la providence ma enviée et qui a partagé avec moi le travail, d'avoir eu la patience et le courage pour achever ce travail, et à tout sa famille.*

A ma famille, mes proches et à ceux qui me donnent de l'amour et de la vivacité. A mes amis qui m'ont toujours encouragé et à qui je souhaité plus de succès.

Chaima

Avec l'expression de ma reconnaissance, je dédie ce modeste travail :

A l'homme mon précieux offre des dieux, qui doit ma vie, ma réussite et tous mon respect : mon cher père ***' Abdelkarim'****.*

*A la femme qui a souffert sans me laisser souffrir, qui n'a jamais dite non à mes exigences et qui n'a épargne aucun effort pour me rendre heureuse : mon adorable mère '****Oumria'****.*

*A mon mari '****Abdelmadjid'*** *qui m'avait toujours soutenu et encourager durant ces années d'étude.*

A mes chères sœurs ' ***Nour Elhouda'****, '****Sara'****, '****Hanene'*** *, '****Rania'*** *et '****Nesrine'****.*

*A '****Karaali Chaima'*** *chère sœur avant d'être binôme pour sa patience et son courage de le long de ce travail et a toute sa famille.*

A ma famille, mes proches et à ceux qui me donnent de l'amour et de la vivacité.

Zineb (Souhila)

Symboles Abréviations

NP	Non déterministe polynomiales
CSOP	Constraint Satisfaction and Optimization Problem
$\mathbb{C}$	L'ensemble des contraintes
D	L'ensemble des domaines
f	Fonction objectif
(X, D, C, f)	L'ensemble des variables du problème.
S	L'ensemble des solutions
s	Solution
X	L'espace naturelle des variables.
$\mathbb{R}$	L'ensemble Réel.
W	L'ensemble des solutions potentielles.
NLP	Non-Linear Programming.
NFL	Théorème "No Free Lunch".
PLNE	Programme Linéaire en Nombres Entiers.
PLM	Programme Linéaire Mixte.
PPC	Programmation Par Contraintes.
CSP	Constraint Search Programming.
CBLS	Constraint Based Local Search.
S_{opt}	Solution optimale.
RL	Recherche Locale.
SA	Simulated Annealing, (recuit simulé).
TS	Tabu Seach, (recherche tabou).
GA	Algorithmes Génétiques (Genetic Algorithm).
ABC	Artificial Bee Colony (la colonie d'abeilles artificielles).
PSO	Particle swarm optimization, (l'optimisation par essaims particulaires
ma	La mère.
pa	Le père.
V	La vitesse.
a	L'accélération.
x	La position.
m	L'indice d'itération.
x_h	La position initiale.
v_i	La vitesse actuelle.
v_h	La vitesse initiale.
x_i	La position actuelle.

P_i	Position personnelle.
P_g	La position globale.
P_h	La Probabilité.
PSOPC	introduit une Congrégation Passive PSO.
PN	La taille de la population.
D	le nombre de dimension.
SN	nombre d'abeilles employées.
LN	nombre d'abeilles spectatrices.

Table des matières

Liste des figures :

Liste des tableaux :

Introduction générale

Introduction générale :

Le succès d'un projet ou le développement d'une entreprise est dans de nombreux cas liés à la capacité des ingénieurs et des décideurs de résoudre des problèmes de type: minimiser les coûts et maximiser la production avec l'inclusion d'un certain nombre de Paramètres. Ce type de problème appelé problème d'optimisation, où nous voulons minimiser une fonction de coût ou maximiser une fonction objective.

Les problèmes d'optimisation apparaissent dans plusieurs domaines, telle que : l'énergie solaire (optimiser la production photovoltaïque, le rendement de hétérojonctions, le dimensionnement des systèmes solaires…), l'analyse numérique (approximation/résolution des systèmes linéaires, non linéaires)…etc. Résoudre un tel problème est de trouver les meilleures valeurs des paramètres pour avoir une solution optimale.

Il y'a généralement deux approches pour résoudre les problèmes d'optimisation : une approche de résolution exacte et approche de résolution approximative. La première cherche à trouver la solution exacte d'un problème donné, comme le branch-and-bound, pratiquement cette approche est limitée à des petits instances de problèmes difficiles. La deuxième tendance est la résolution approximative, où nous essayons de trouver des bonnes solutions en utilisant des méthodes stochastiques appelées méta-heuristiques.

Les méta-heuristiques sont souvent inspirées à partir des systèmes naturels, qu'ils soient pris en biologie de l'évolution (cas des algorithmes génétique(GA)), ou encore en éthologie (cas des algorithmes de colonie des abeilles(ABC) ou de l'optimisation par essaim de particules(PSO)).

Notre travail est représenté sous trois chapitres, le premier donne une aperçu générale sur les notions liées à l'optimisation (définition, classification et exemples).

Dans le deuxième chapitre, on va discuter les méthodes de résolution des problèmes d'optimisation par méta-heuristique.

Le troisième chapitre est consacré pour l'étude comparative de trois algorithmes méta-heuristiques choisis (GA, PSO et ABC).

Chapitre I :

Généralité sur les méthodes d’optimisation

1.1 Introduction :

La programmation mathématique traite des problèmes d'optimisation qui posséder une structure spécifique : Maximiser (ou minimiser) une fonction objectif soumis à un ensemble de contraintes qui définissent la faisabilité. La fonction objective et les contraintes sont des fonctions mathématiques de variables de décision et paramètres.
Les variables de décision sont les aspects d'un système qui sont contrôlables, tandis que les paramètres sont des quantités données qui ne sont pas contrôlables. En structurant et en résolvant un programme mathématique, l'analyste tente de découvrir les décisions concernant le système étudié qui sont, dans un certain sens, les meilleures.
Une collection de valeurs pour chacune des variables de décision est appelée une solution au programme mathématique. Une solution qui satisfait toutes les contraintes est appelée une solution réalisable. En général, il existe un nombre énorme de solutions réalisables. Le rôle de la fonction objectif est de fournir une base pour l'évaluation des solutions réalisables. Cette solution réalisable qui donne la meilleure valeur (la plus basse ou la plus élevée) de la fonction objectif est appelée la solution optimale [1].
D'après certains dictionnaires, l'usage français du verbe optimiser nous est arrivé vers 1844 d'Angleterre, ou « to optimiste » signifiait « se comporter en optimise » ; on peut donc dire que l'optimiseur est comme l'optimiste qui pense toujours mieux faire : « minimiser un cout » , « maximiser un profit », « optimiser un procédé », « gagner en optimisant » ,… etc. autant d'expression qui font référence à un domaine encore relativement jeune des mathématiques et de leurs applications : l'optimisation [2].

1.2 Définition d'optimisation :

Une fois que l'on dispose d'un outil pour évaluer le comportement et les performances d'un système étudié, on cherche à optimiser son fonctionnement. Il peut s'agir d'une étude monocritère, comme la minimisation du coût ou de la durée, ou d'une étude multi-objectifs dans laquelle on recherche le meilleur compromis entre différents indicateurs de performance comme la minimisation d'un cout [3].
Les ingénieurs et les décideurs sont confrontés quotidiennement à des problèmes de complexité grandissante, qui surgissent dans des secteurs techniques très divers, comme dans la conception de systèmes mécaniques, le traitement des images, l'électronique ou la recherche opérationnelle. Le problème à résoudre peut souvent s'exprimer comme un problème d'optimisation : on définit une fonction objectif, ou fonction de coût (voire plusieurs), que l'on cherche à minimiser ou à maximiser par rapport à tous les paramètres concernés. La définition du problème d'optimisation est souvent complétée par la donnée de contraintes. On distingue les contraintes impératives (ou « hard ») et les contraintes indicatives (ou « soft»). Tous les paramètres des solutions retenues doivent respecter les contraintes du premier type, qui peuvent aussi être vues comme définissant l'espace de recherche, faute de quoi ces solutions ne sont pas réalisables. Les contraintes indicatives doivent simplement être respectées aussi bien que possible [4].
Les problèmes d'optimisation que l'on rencontre dans l'étude des systèmes optimaux sont souvent combinatoires. Ce type de problème se caractérisent par le fais que plus le nombre de solutions (ou combinaisons) possibles augmente considérablement (exponentiellement) plus la taille du problème augmente. Par conséquent, le temps de calcul nécessaire à l'énumération de toutes les combinaisons s'en trouve directement affecté. Un grand nombre de ces problèmes d'optimisations sont NP-difficiles. Cela signifie que si l'on veut trouver la solution optimale d'un tel problème, il est nécessaire d'énumérer toutes les solutions possibles et que cette énumération nécessite un temps de calcul qui augmente exponentiellement avec la taille du problème. Or, les outils d'aide à la décision que l'on souhaite utiliser pour résoudre des

problématiques issues de systèmes doivent donner une réponse rapide, et le meilleur possible. Ceci nous amène donc à classer les méthodes d'optimisation en différentes catégories : les méthodes d'optimisation exactes qui sont couteuses en temps de calcul mais qui garantissent l'optimalité de la solution et les méthodes approchées (heuristique et méta-heuristique) qui sont plus rapides mais qui ne garantissent pas l'optimalité de la solution obtenue.
Dans le milieu de la conception, l'optimisation est le fait d'optimiser une fonction.

Formulation générale d'une fonction à optimiser :

$$f : R^n \to R : x \to f(x) \text{ avec } \begin{cases} n\, x \in R & \begin{cases} g(x) \le 0 \\ h(x) = 0 \end{cases} \\ \text{possibilité de contraintes} & \end{cases} \quad (0.1)$$

Formulation mathématique de l'optimisation :

$$\min f(x) \text{ tel que } x \in C \quad (0.2)$$

Avec C, l'ensemble des paramètres respectant les contraintes [5].

L'optimisation intervient dans de nombreux domaines :

- ✓ En recherche opérationnelle (e.g., problème de transport, économie…etc).
- ✓ En analyse numérique (e.g., approximation/résolution des systèmes linéaires, non linéaire).
- ✓ En automatique (e.g., modélisation de système, filtrage…etc).
- ✓ En énergie solaire (e.g., optimiser la production photovoltaïque, le rendement de hétérojonctions, le dimensionnement des systèmes solaires…).

1.3 Quelques définitions :

1.3.1 Algorithmes en temps polynomial :

Un algorithme est dit en temps polynomial s'il existe un polynôme $Q(n)$ en n majorant $T(n)$ pour toute donnée numérique [3] .

$$T(n) \le Q(n)$$

1.3.2 Algorithmes pseudo-polynomiaux en temps :

Un algorithme est dit de temps pseudo-polynomial s'il n'est pas de temps polynomial, mais il existe un polynôme qui dépend de la taille du problème n et une autre grandeur caractéristique P telle que :

$$T(n) \le Q(n, P)$$

1.3.3 Problèmes NP-difficiles :

Dans un problème d'optimisation, on cherche le vecteur x dans l'ensemble des solutions X tel que la fonction $f(x)$ soit minimisée. On peut définir un problème de décision associé à un problème d'optimisation comme, par exemple, existe-t-il un vecteur x dans X tel que $f(x) \prec A$? La réponse à un tel problème sera oui ou non. Les problèmes de décision qui ne peuvent pas être résolus en temps polynomial sont dits NP-complets .Le problème d'optimisation correspondant est alors considéré comme NP-difficile [3].

1.3.4 Extremum d'une fonction définie sur un intervalle I de $\mathbb{R}$

Soit $f: \to \mathbb{R}$ une fonction définie sur un intervalle I et soit $a \in I$.

On dit que f admet un **maximum** en a ,Si, pour tout $x \in I$, $f(x) \geq f(a)$.

On dit que f admet un **minimum** en a ,Si, pour tout $x \in I$, $f(x) \leq f(a)$.

On parle parfois de maximum ou de minimum **global** (Figure 1.1) de la fonction, et on dit que $f(a)$ est le maximum (resp. le minimum) de f sur I. On dit aussi que m est un **extremum** de f si c'est un maximum ou un minimum.

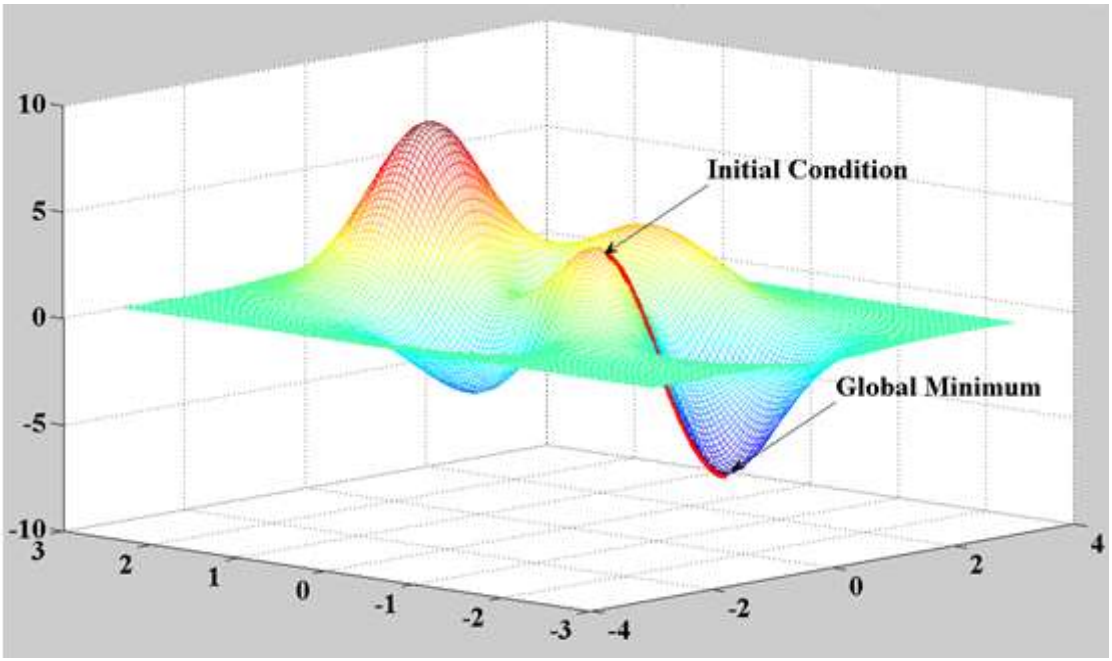


Figure 1.1 : exemple d'un minimum global [6].

On dit que f admet un maximum local (ou relatif) en a s'il existe un intervalle ouvert j contenant a tel que, pour tout $x \in J \cap I$, on a $f(x) \leq f(a)$. On définit de même un minimum local en inversant le sens de l'inégalité. Un extremum local est un maximum ou un minimum local (Figure 1.2).

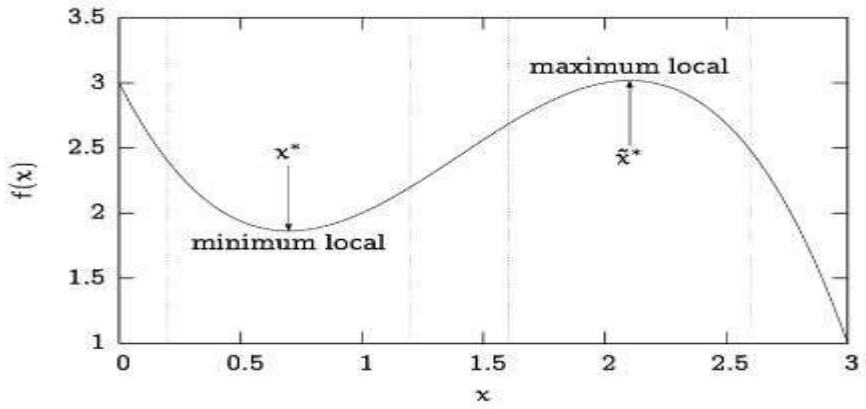


Figure 1.2 : exemple d'un minimum et maximum local [6].

1.4 Les différentes natures des problèmes d'optimisations :

Les problèmes d'optimisation sont définis selon plusieurs caractéristiques :

- ✓ Optimisation combinatoire
- ✓ Différentiable ou non différentiable
- ✓ Mono-objectif ou multi-objectif
- ✓ Linéaire ou non-linéaire
- ✓ Contraint ou non-contraint… Etc.
- ✓ Cela implique qu'un outil d'optimisation peut être développé pour résoudre un/plusieurs type(s) de problème ou être indépendant du problème [5].

1.4.1 Optimisations Combinatoires :

L'optimisation combinatoire est le domaine des mathématiques discrètes qui traite de la résolution du problème suivant :

Soit X un ensemble de solutions admissibles. Soit X une fonction permettant d'évaluer chaque solution admissible. Il s'agit de déterminer une solution s appartenant à X qui minimise f. L'ensemble X des solutions admissibles est supposé fini et est en général défini par un ensemble $\mathbb{C}$ de contraintes. Malgré l'évolution permanente des calculateurs et les progrès fulgurants de l'informatique, il existera certainement toujours, pour un problème (P) difficile, une taille critique de X au-dessus de laquelle même une énumération partielle des solutions admissibles devient prohibitive. Compte tenu de ces difficultés, la plupart des spécialistes de l'optimisation combinatoire ont orienté leur recherche vers le développement de méthodes heuristiques. Une méthode heuristique est souvent définie comme une procédure exploitant au mieux la structure du problème considéré, dans le but de trouver une solution de qualité raisonnable en un temps de calcul aussi faible que possible [7].

L'objectif en optimisation combinatoire, est de choisi parmi l'ensemble des configurations possibles pour le problème qu'on a à traiter, une configuration qui satisfasse les contraintes et optimise les critères. Ces critères sont généralement représentés par une fonction appelée fonction cout ou objectif (fitness) [8].

1.4.2 Optimisations différentiable :

Soit$f: \mathbb{R}^n \to \mathbb{R}$ une fonction continue. Si, pour tout$d \in \mathbb{R}^n$, la dérivée directionnelle de f dans la direction d existe, alors la fonction f est dite différentiable.

1.4.3 Optimisation non différentiable :

C'est le vocable sous lequel on répertorie les problèmes d'optimisation où certaines données-fonctions ne sont pas (partout) différentiables. Cela se produit plus fréquemment qu'on l'imagine ; et même les si la zone ou les données ne sont pas différentiables ne concernent que « très peu » de points, ce sont souvent ces points qui intéressent l'optimiseur (ils sont en fait incontournables). A titre d'exemple, considérons une fonction-objectif f de la forme :

$$f(x) := max\{ f_1(x), \ldots, f_m(x) \} \quad (0.3)$$

Où les fonctions f_i sont différentiables (c'est une forme de critères intervenant en approximation ou économie mathématique [2].

1.4.4 Optimisations linéaires :

La programmation linéaire constitue l'une des acquisitions plus importantes de la théorie économique d'après la deuxième guerre mondiale. Elle s'est développée très rapidement, grâce aux efforts conjugués des mathématiciens, des chefs d'entreprises, des chefs militaires, des statisticiens et des économistes. Les problèmes de la programmation linéaire 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. Soulignons à ce propos, qu'une contrainte est linéaire, lorsqu'elle s'exprime par une égalité ou inégalité dont le premier membre est une combinaison linéaire et le second, un nombre réel, le programme suivant est de ce type [9]:

Trouver $y_1 0,\ y_2 0,...,y_i 0,\ ...,\ 0$ tel que :

$$a_{11}\,y_1 + a_{12}\,y_2 + ... + a_{1i}\,y_i + ... + a_{1n}\,y_n\ q_1$$

$$a_{21}\,y_1 + a_{22}\,y_2 + ... + a_{2i}\,y_i + ... + a_{2n}\,y_n\ q_2$$

$$a_{m1}\,y_1 + a_{m2}\,y_2 + ... + a_{mi}\,y_i + ... + a_{mn}\,y_n\ q_m$$

Et rendant

$$P_1\,y_1 + P_2\,y_2 + ... + P_j\,y_j + ... + P_n\,y_n \quad \text{maximum.} \qquad (0.4)$$

C'est une traduction littérale de « linear- programming », irrécupérable à présent ; optimisation a données linéaires (ou affines) serait plus appropriée (Figure1.3). Il s'agit d'une classe de problèmes d'optimisations ou la fonction objectif est linéaire, $f(x) := \langle c, x \rangle = \sum_{i=1}^{n} c_i x_i$ et ou l'ensemble contrainte C est un polyèdre convexe fermé de$\mathbb{R}^n$. C'est un domaine qui a connu un développement extraordinaire au sortir de la deuxième guerre mondiale, et qui reste en haut de l'affiche en optimisation depuis plus de cinquante ans. Il y a de nombreux résultats mathématiques et des algorithmes de résolutions qui sont spécifiques à ce « monde linéaire » [2].

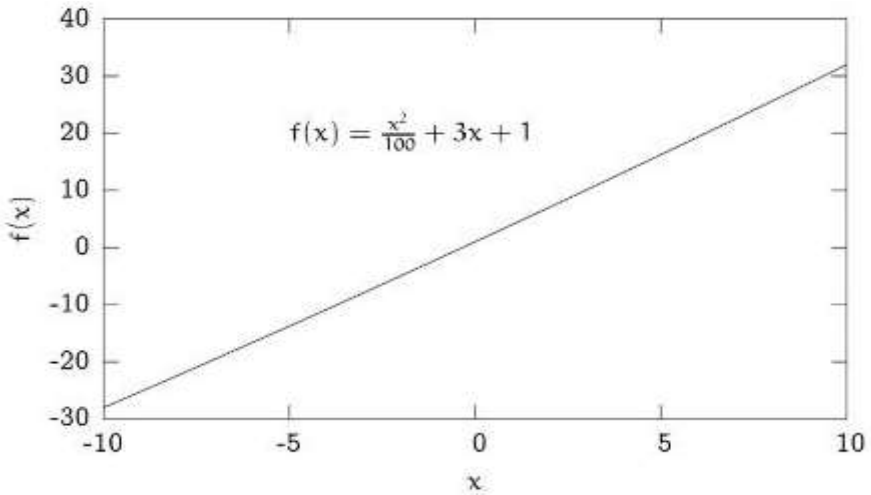


Figure 1.3 : exemple d'une fonction presque linéaire [6].

1.4.5 Problèmes d'optimisation non linéaires :

l'optimisation non linéaire (en anglais : non-linear programming – NLP) s'occupe principalement des problèmes d'optimisation dont les données, i.e., les fonctions et ensembles définissant ces problèmes, sont non linéaires, mais sont aussi différentiables autant de fois que nécessaire pour l'établissement des outils théoriques, comme les conditions d'optimalité, ou pour la bonne marche des algorithmes de résolution qui y sont introduits et analysés (Figure 1.4).

On a une fonction $f : X \to R$, avec $X \subseteq R^n$. L'objectif est de déterminer le vecteur x défini par :

$$x \in X ; f(x) = \min_{y \in X} f(y). \quad (0.5)$$

De façon équivalente, on peut rechercher la valeur pour laquelle *f* est maximale :

$$x \in X ; f(x) = \max_{y \in X} f(y). \quad (0.6)$$

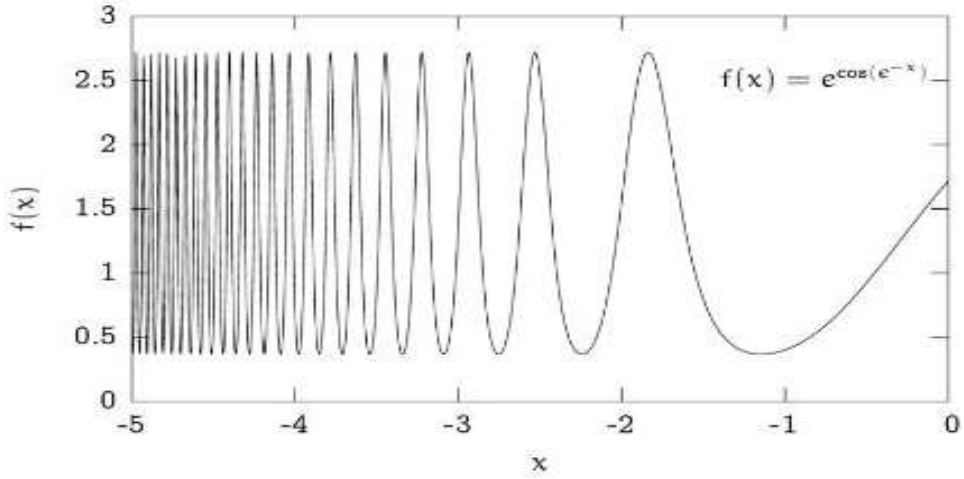


Figure 1.4 : exemple d'une fonction fortement non-linéaire [6].

1.4.6 Problèmes d'optimisation sous contraintes :

Beaucoup de problèmes combinatoires peuvent également s'exprimer comme des problèmes d'optimisation sous contraintes (Constraint Satisfaction and Optimization Problem ou CSOP). Un problème d'optimisation sous contraintes est un problème pour lequel on cherche parmi l'ensemble de toutes les solutions réalisables la meilleure solution selon une fonction qui définit un objectif donné. La fonction objective pour rôle d'évaluer la qualité d'une solution tandis que les contraintes permettent d'éliminer les configurations qui ne sont pas des solutions. Le problème est alors double : le premier consiste à trouver l'ensemble de toutes les solutions réalisables et le deuxième à trouver dans cet ensemble la meilleure solution qui minimise ou maximise la fonction objective. On parle d'un problème de minimisation quand la qualité d'une solution est donnée par une fonction objectif à minimiser et d'un problème de maximisation quand la qualité d'une solution est donnée par une fonction objectif à maximiser. L'étude d'un problème d'optimisation révèle habituellement des contraintes impératives (structurelles ou dures) et des contraintes indicatives (préférences ou molles) [10].Les solutions retenues doivent respecter les contraintes impératives qui limitent l'espace de recherche tandis que les

contraintes indicatives doivent être respectées autant que possible, le niveau de satisfaction de ces dernières devant être encodé dans la fonction objectif.

(CSOP). Un problème d'optimisation sous contraintes (CSOP) est défini par le quadruplet.

(X, D, C, f) *avec* : $X = \{x_1, \cdots, x_n\}$ C'est l'ensemble des variables du problème,

- $D = \{D(x_1), \cdots, D(x_n)\}$ est l'ensemble des domaines,
- $C = \{C_1, C_2, ..., C_k\}$ représente l'ensemble des contraintes,
- f est une fonction objective définie sur un sous-ensemble de X (à minimiser ou à maximiser).

La fonction f permet de définir une relation d'ordre total entre n'importe quelle paire de solutions dans S . Nous avons vu à la section qu'on pouvait associer un problème de décision à chaque problème d'optimisation, le but étant de déterminer s'il existe une solution pour laquelle la fonction objectif soit supérieure (respectivement inférieure) ou égale à une valeur donnée. Ainsi, la complexité d'un problème d'optimisation est liée à celle du problème de décision qui lui est associé. En particulier, si le problème de décision est NP-complet, alors le problème d'optimisation est NP-difficile [11].

1.4.7 Optimisation multicritères (ou multi objectif) :

Pour certains problèmes combinatoires, plusieurs objectifs contradictoires doivent être optimisés en même temps. Par exemple, lors de la préparation d'un voyage, il peut être demandé de choisir entre différents modes de déplacement (avion, train, bateau, voiture, etc.) de façon à minimiser le coût de transport tout en minimisant le temps du trajet.

On recense essentiellement deux approches associées à ces problèmes [1] : les techniques de génération(que nous appellerons multi-objectif) et les techniques à base de préférences (que nous nommerons multicritère). Les techniques basées sur les préférences consistent à synthétiser les informations relatives à différents points de vue ou aspects. Il s'agit de ramener un problème multi-objectif à un problème simple-objectif. Elles utilisent une méthode pour classer les objectifs et ainsi trouver une solution qui optimise le classement. Ce classement peut être réalisé par une variété de moyens allant d'une simple somme pondérée des objectifs à une fonction d'utilité complexe [12, 13]. Les techniques de génération quant à elles recherchent dans un ensemble très grand de solutions celles qui satisfont au mieux l'ensemble des objectifs. Dans un contexte d'aide à la décision, les problèmes contiennent souvent une multitude d'aspects simultanés prendre en considération pour la prise de décision. Pour S. BEN-MENA [14], l'approche multicritère, capable d'intégrer tout type de critères, permet alors de se diriger vers un compromis judicieux plutôt qu'un optimum qui n'a pas toujours de sens. Une technique classique consiste alors à encoder chaque aspect du problème dans la fonction objectif, en attribuant à chaque critère un éventuel poids pour lui accorder plus ou moins d'importance par rapport aux autres critères [15]. Bien que cette approche du « critère unique de synthèse » soit soumise à une compensation possible entre critères ou à une forte sensibilité aux changements d'échelle, les méthodes d'agrégation complète peuvent s'avérer intéressantes ou tout simplement les seules utilisables [16].

1.4.8 Les problèmes d'optimisation mono-objectifs :

Lorsqu'un seul objectif (critère) est donné, le problème d'optimisation est mono-objectif. Dans ce cas la solution optimale est clairement définie, c'est celle qui a le coût optimal (minimal, maximal). De manière formelle, à chaque instance d'un tel problème est associé un ensemble W des solutions potentielles respectant certaines contraintes et une fonction d'objectif $W \in Y$ qui associe à chaque solution admissible S de W une valeur (S). Résoudre l'instance (W) du problème d'optimisation consiste à trouver la solution optimale S^* de W qui optimise (minimise ou maximise) la valeur de la fonction objective pour le cas de la minimisation : le but est de trouver S^* de W tel que (S^*)ò(S) pour tout élément S de W. Un problème de maximisation peut être défini de manière similaire.

L'optimisation du problème mono-objectif peut garantir l'optimalité de la solution trouvée, mais n'en trouve qu'une seule, dans les situations réelles, les décideurs ont généralement besoin de plusieurs alternatives.

1.5 Les méthodes de résolution des problèmes d'optimisations difficiles :

Les problèmes d'optimisation combinatoire constituent une catégorie de problèmes très difficiles à résoudre [17]. La résolution d'un problème d'optimisation combinatoire consiste à explorer un espace de recherche afin de minimiser ou maximiser une fonction objectif pour trouver la meilleure solution, définie comme la solution globalement optimale ou optimum global. Cette résolution est habituellement délicate puisque le nombre de solutions réalisables croît généralement avec la taille du problème, rendant un parcours exhaustif impraticable en pratique.

La complexité en taille ou en structure de l'espace de recherche d'une part et de la fonction objectif d'autre part peut conduire à utiliser des méthodes de résolution différentes en fonction du problème ou de l'instance à traiter. Intuitivement, on pourrait penser que la difficulté d'une instance d'un problème est proportionnelle au nombre de variables, à la taille du domaine par variable et au nombre de contraintes. Dans les faits, il est le plus souvent impossible de prévoir avec certitude l'efficacité d'une méthode donnée pour un problème particulier. Le théorème "no free lunch" ("pas de dîner gratuit, NFL") [18] nous indique, sous certaines hypothèses, qu'une méthode de résolution ne peut prétendre être la plus efficace sur tous les problèmes. Les méthodes de résolution liées aux problèmes d'optimisation peuvent être regroupées en deux grandes familles (figure1.5) :

- ✓ **les méthodes dites complètes (ou exactes)** qui obtiennent les solutions optimales et garantissent les résultats ;
- ✓ **les méthodes dites incomplètes (ou approchées)** qui trouve rapidement des solutions de bonne qualité sans garantir l'optimalité.

L'optimisation combinatoire couvre un large éventail de techniques et fait toujours l'objet de recherches intensives sur de nombreux domaines d'application (e.g., finance, transport, surveillance, biologie, . . . etc.) de sorte qu'il est impossible de toutes les aborder ici en détail. Nous nous limitons dans un premier temps à un panorama non exhaustif de chaque famille pour avoir une idée très générale des méthodes existantes.

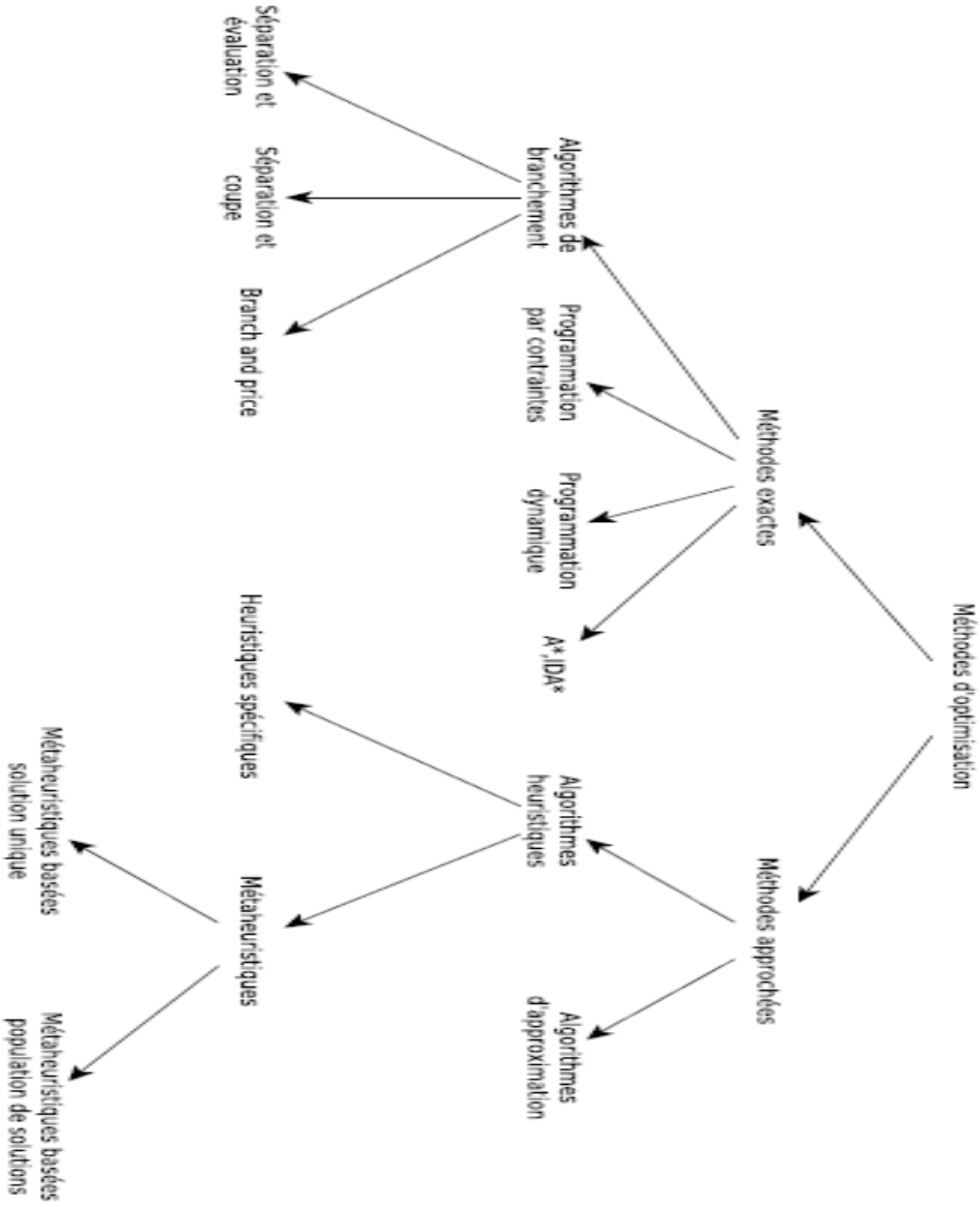


Figure 1.5 : Méthodes classiques d'optimisation [8].

1.5.1 Les méthodes exactes (complètes) :

Les algorithmes complets (ou exacts) évaluent l'espace de recherche dans sa totalité en réalisant une énumération intelligente. Ils donnent la garantie de trouver la solution optimale pour une instance de taille finie dans un temps limité et permettent de prouver son optimalité [19].

1.5.1.1 La programmation mathématique :

En programmation mathématique, l'objectif du problème étudié est exprimé sous la forme d'une fonction d'une variable (appelée variables de décision).Cette fonction objectif est soumise à un ensemble de contraintes d'égalité ou d'inégalité, qui délimitent l'espace des solutions à explorer. Lorsque les variables de décision sont des nombres entiers et que les fonctions sont linéaires, il s'agit d'un modèle de programmation linéaire en nombres entiers. Dans le cas des variables entières réelles, le modèle se situe dans le domaine de la programmation entière mixte. On rencontre aussi des problèmes de programmation binaire et de programmation non linéaire.
Différents outils existent pour obtenir la solution optimale au problème à partir du modèle mathématique. Les exemples incluent la méthode du simplexe, des solveurs dédiés tels que CPLEX, XPRESS, GAMS et EXCEL [3].

1.5.1.2 La programmation linéaire :

La programmation linéaire propose un cadre de modélisation mathématique permettant de résoudre des problèmes d'optimisation avec des algorithmes efficaces comme la méthode du simplexe [20]ou les méthodes de points intérieurs [21]. On parle de programme linéaire en nombres entiers (PLNE) lorsque les valeurs des variables du programme linéaire doivent être des nombres entiers et de programme linéaire mixte (PLM) lorsque certaines variables doivent être entières et d'autres pas. Relaxer un PLNE ou un PLM consiste en général à supprimer certaines de ses contraintes ou à les remplacer par des contraintes plus faibles. Le problème résultant est souvent plus simple à résoudre que le problème original et sa solution optimale peut être soit réalisable pour le problème original, donc optimale pour ce dernier, soit non réalisable et avoir un coût proche de l'optimum du problème original.

1.5.1.3 L'algorithme de séparation et évaluation :

L'algorithme de séparation et évaluation (Branch and Bound) [26] utilise une méthode arborescente de recherche basée sur l'idée d'énumérer implicitement les solutions possibles d'un problème de PLNE pour trouver la meilleure solution sans passer par une énumération explicite trop gourmande en temps. Cet algorithme est composé de trois éléments principaux :

- ✓ **la séparation :** consiste à diviser le problème en sous-problèmes où chaque sous-problème contient un ensemble de solutions réalisables tel que l'union de leurs espaces de solutions forme l'espace des solutions du problème père. Cela revient à construire un arbre (arbre de recherche ou arbre de décision) permettant d'énumérer toutes les solutions. Ainsi, en résolvant tous les sous-problèmes et en gardant la meilleure solution trouvée, on est assuré d'avoir résolu le problème initial.
- ✓ **l'évaluation** : permet de réduire l'espace de recherche en éliminant les sous-ensembles qui ne contiennent pas la solution optimale. Le principe consiste à mémoriser la solution de plus bas coût (pour un problème de minimisation) rencontrée pendant l'exploration et à comparer le coût de chaque nœud parcouru avec celui de la meilleure solution. Si le coût du

nœud considéré est supérieur au meilleur coût trouvé, on peut arrêter l'exploration de la branche car toutes les solutions de cette branche seront nécessairement moins bonnes.

- ✓ **une stratégie de parcours :** en largeur, en profondeur ou le meilleur d'abord

1.5.1.4 La programmation dynamique

La programmation dynamique repose sur le principe d'optimalité de BELLMAN [28]selon lequel une séquence optimale est composée de sous-séquences optimales. La solution optimale d'un problème peut alors être déduite en combinant des solutions optimales d'une série de sous-problèmes. On commence par les sous-problèmes les plus petits et on remonte vers les sous-problèmes de plus en plus difficiles en tirant profit des résultats des problèmes déjà obtenus. La difficulté réside dans la formulation d'une séquence de décisions permettant de calculer la solution du problème en fonction des solutions des sous-problèmes.

Son efficacité dépend de la propriété récursive et de l'efficacité des procédures de résolution des problèmes intermédiaires.

1.5.1.5 La programmation par contraintes

Dans l'absolu, la programmation par contraintes (PPC) se situe au même niveau qu'un CSP et englobe à la fois des méthodes de résolution exactes et approchées (e.g., Constraint Based Local Search [29],CBLS). Par abus de langage, nous réduisons la PPC au champ des méthodes de résolution complètes.

La programmation par contraintes (Constraint Programming) [30],[31],[32], [33] est une approche très populaire qui repose sur le formalisme CSP avec une recherche arborescente incluant des notions liées à la consistance locale des contraintes : elle effectue une recherche énumérative complète en inférant des réductions de l'espace de recherche à partir des contraintes du problème. La recherche réalise un parcours de l'espace de recherche par un jeu d'affectations et de retours arrière [34]. Souvent représentée par un arbre, la recherche énumère toutes les instanciations possibles jusqu'à trouver une solution ou conclure qu'il n'y en a pas. La propagation quant à elle vient assister la recherche en essayant de déduire de nouvelles informations à partir de l'état courant des domaines. À partir d'une instanciation partielle, le mécanisme de propagation tente de supprimer des valeurs impossibles des domaines des variables, réduisant ainsi la taille de l'espace de recherche.

1.5.2 Panorama des méthodes approchées (incomplètes) :

Les méthodes incomplètes (approchées) reposent en grande partie sur l'utilisation d'heuristiques [35], [36]. Elles explorent une sous-partie de l'espace de recherche au moyen de techniques variées afin d'en extraire au plus vite une solution qu'on espère de bonne qualité. Contrairement aux méthodes exactes (complètes), celles-ci ne sont pas en mesure de prouver l'optimalité des solutions trouvées. L'efficacité d'une méthode incomplète réside dans l'équilibre entre deux grandes notions : l'intensification d'une part et la diversification d'autre part. L'intensification consiste à scruter une zone précise de l'espace de recherche afin d'en extraire un optimum local (et si possible l'optimum global) alors que la diversification permet de déplacer la recherche dans des zones variées et prometteuses de l'espace de recherche non encore explorées. Dans la classe des algorithmes approchés, on peut distinguer trois sous-classes :

• les algorithmes d'approximation ;

• les heuristiques ;
• les méta-heuristiques ;

1.5.2.1 Les algorithmes d'approximation :

L'approximation d'un problème d'optimisation NP-difficile consiste à obtenir des informations sur sa solution optimale sans la connaître. En général, il s'agit de résoudre un problème plus simple que le problème original, pour transformer ensuite la solution optimale S_{opt} du problème simplifié en une solution du problème original (voir la figure 1.6). La difficulté est de garantir une distance maximale entre l'approximation S et l'optimum S_{opt} du problème original. Le problème simplifié est souvent obtenu en relâchant des contraintes du problème original. Plus formellement, un algorithme d'approximation est un algorithme polynomial qui renvoie une solution approchée garantie au pire cas à un facteur de la solution optimale [37]. Parfois, le facteur d'approximation dépend de la taille de l'instance du problème. Toutefois, pour certains problèmes NP-difficile, il est impossible de faire des approximations. De plus, ces algorithmes sont spécifiques au problème ciblé et les approximations fournies sont souvent trop éloignées de la solution optimale, ce qui freine leur utilisation pour beaucoup d'applications réelles [8].

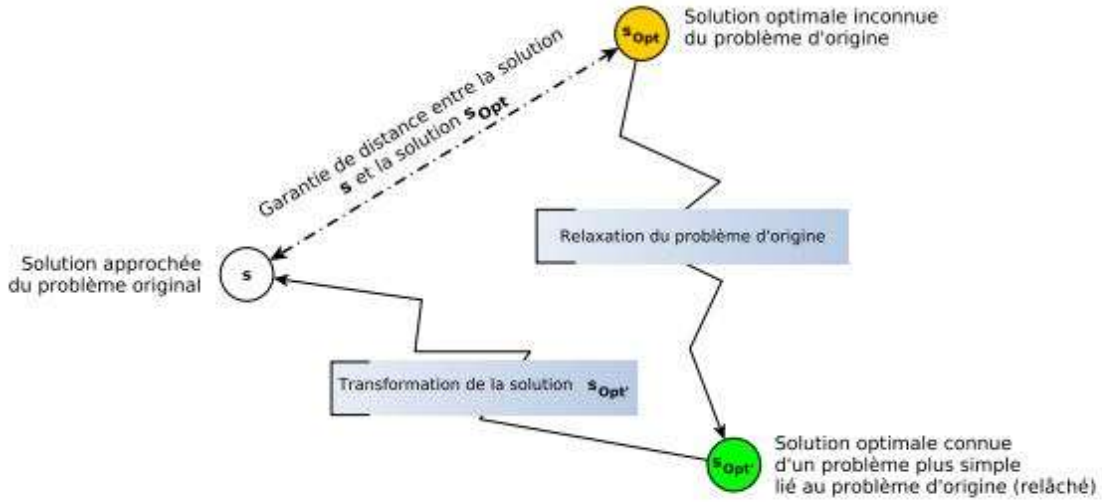


Figure 1.6 : Principe lié aux algorithmes d'approximation [38].

1.5.2.2 Les heuristiques

Le terme heuristique peut avoir différentes significations et être employé dans différents contextes, y compris par exemple dans le cadre des méthodes exactes pour le choix d'une variable ou d'une valeur particulière à évaluer (e.g., dans un arbre de décision). Nous l'employons ici pour représenter trois notions distinctes :

• soit une classe de méthodes approchées d'optimisation conformément à la classification proposée dans la figure1.5. Dans ce cas, les méta-heuristiques représentent une déclinaison de cette famille ;

• soit des méthodes d'optimisation gloutonnes capables de construire une solution en partant d'une configuration initialement vide et en choisissant à chaque étape la meilleure composante possible insérer dans la solution (par exemple une variable de décision et une valeur de son domaine) sans jamais remettre en cause cette décision par la suite.

• soit des algorithmes problèmes-dépendants (spécifiques) connectés aux méta-heuristiques pour fournir une implémentation concrète aux méthodes d'optimisation abstraites, la partie abstraite étant représentée par les méta-heuristiques.

1.5.2.3 Les méta-heuristiques :

Dans certaines situations, on doit chercher en un temps raisonnable des solutions de bonne qualité sans pour autant garantir l'optimalité. Les méta-heuristiques contournent le problème d'explosion combinatoire en n'exploitant délibérément qu'une partie des combinaisons prometteuses. Par conséquent, elles peuvent ne pas trouver la solution optimale, et encore moins prouver l'optimalité de la solution trouvée. En général, elles n'offrent pas non plus de garantie d'approximation. En contrepartie, elles permettent d'obtenir rapidement de bonnes solutions pour les problèmes de grandes tailles ou très difficiles. Certaines méta-heuristiques partent d'une solution valide qu'elles modifient légèrement à chaque itération dans le but de l'améliorer. Au lieu de compter sur une solution unique pour découvrir de meilleures solutions, certaines méta-heuristiques utilisent plusieurs solutions de départ. Elles combinent alors les propriétés de différentes solutions en espérant trouver des solutions de meilleure qualité. Alors qu'une heuristique est souvent définie comme une procédure spécifique exploitant au mieux la structure d'un problème dans le but de trouver une solution de qualité raisonnable en un temps de calcul aussi faible que possible [39], les méta-heuristiques sont des algorithmes génériques qui peuvent être utilisés pour résoudre une grande variété de problèmes d'optimisation et d'instances différentes. Elles peuvent être vues comme des méthodes générales de niveau supérieur (méta) utilisées pour guider la stratégie de conception d'heuristiques sous-jacentes, ces dernières étant adaptées au problème d'optimisation à résoudre. On retrouve au niveau des méta-heuristiques les méthodes basées sur une solution unique qui manipulent et transforment une seule solution pendant le processus de recherche tandis que les algorithmes à base de population travaillent avec une population entière, c'est à dire un ensemble de solutions qui évolue au fur et à mesure des itérations [8],[40].

- ✓ **Méta-heuristiques basées sur une solution unique :**

Les méta-heuristiques basées sur une solution unique regroupent les méthodes dites de recherche locale (ou à base de voisinages). Les méthodes de recherche locale (RL) sont des algorithmes itératifs qui, à partir d'une solution de départ complète, explorent l'espace de recherche en se déplaçant pas à pas d'une solution à une autre. Dans cette catégorie, nous pouvons citer parmi les plus populaires les méthodes de le recuit simulé (Simulated Annealing, SA) et la recherche tabou (Tabu Seach, TS) [41].

- ✓ **Méta-heuristiques basées sur une population de solutions :**

Les méta-heuristiques basées sur une population de solutions utilisent la population comme facteur de diversité et font intervenir plusieurs stratégies d'évolution de cette population conduisant à des méthodes telles que les algorithmes génétiques (Genetic Algorithm), la recherche par dispersion (Scatter Search, SS) ,l'algorithme des colonies de fourmis (Ant Colony Optimization, ACO) ou encore l'optimisation par essaims particulaires (Particle swarm optimization, PSO) [41].

1.6 Conclusion :

Ce chapitre résume les différents types de problèmes d'optimisation qui existent et les différentes méthodes de résolution de ces dernières. Il existe deux méthodes de résolution, les méthodes dites complètes et les méthodes approchées ou on va s'intéresser aux méthodes incomplètes (approchées) plus précisément les méta-heuristiques dans le prochain chapitre.

Chapitre II :

Les méta-heuristiques :

2.1 Introduction :

L'optimisation combinatoire [36]occupe une place très importante en recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation [42]et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire [43].Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles et ne possèdent donc pas à ce jour de solution algorithmique efficace valable pour toutes les données [44].
Il est donc souvent nécessaire de trouver des modes de résolution qui fournissent une solution de bonne qualité dans un laps de temps raisonnable [45]: c'est ce que font les heuristiques. Depuis une vingtaine d'années, les heuristiques les plus populaires, et également les plus efficaces, sont des techniques générales, appelées méta-heuristiques, qu'il s'agit d'adapter à chaque problème particulier. Parmi ces techniques générales, nous nous pencherons dans ce chapitre sur deux approches ayant clairement démontré leur utilité dans de nombreux domaines, y compris la gestion des systèmes de production. La première de ces approches, appelée Recherche Locale, est couramment utilisée depuis plus de 20 ans et comprend, entre autres, la technique du Recuit Simulé [46] et la Recherche Taboue [47]. La deuxième approche, appelée Méthode Évolutive, est plus récente puisqu' elle date du début des années 90. Elle s'est particulièrement fait connaître par l'intermédiaire des algorithmes génétiques dont l'origine remonte aux travaux de Holland [48].
Dans ce chapitre, on va parler sur Les concepts de base des algorithmes d'optimisation méta-heuristiques et la formulation du problème d'optimisation générique correspondant sont introduits. Principalement, l'algorithme génétique (GA), l'optimisation particulière des essaims (PSO) et la colonie d'abeilles artificielles (ABC) sont pris en compte.

2.2 Généralités sur les méta-heuristiques :

2.2.1 Propriétés :

Une méta-heuristique est constituée d'un ensemble de concepts fondamentaux qui permet d'aider à la conception des méthodes heuristiques pour un problème d'optimisation. Ainsi, une méta-heuristique est adaptable et applicable à une large classe de problèmes. Un ensemble de propriétés intéressantes a été proposé qui caractérisent les méta-heuristiques [49]:

- ✓ Les méta-heuristiques sont des stratégies qui permettent de guider la recherche d'une solution optimale.
- ✓ Le but visé par les méta-heuristiques est d'explorer l'espace de recherche efficacement afin de déterminer des solutions (presque) optimales.
- ✓ Les techniques qui constituent des algorithmes de type méta-heuristique vont de la simple procédure de recherche locale à des processus d'apprentissage complexes.
- ✓ Les méta-heuristiques sont en général non-déterministes et ne donnent aucune garantie d'optimalité.
- ✓ Les méta-heuristiques peuvent contenir des mécanismes qui permettent d'éviter d'être bloqué dans des régions de l'espace de recherche.

- ✓ Les concepts de base des méta-heuristiques peuvent être décrits de manière abstraite, sans faire référence à un problème spécifique.
- ✓ Les méta-heuristiques peuvent faire appel à des heuristiques qui tiennent compte de la spécificité du problème traité mais ces heuristiques sont contrôlées par une stratégie de niveau supérieur.
- ✓ Les méta-heuristiques peuvent faire usage de l'expérience accumulée durant la recherche de l'optimum pour mieux guider la suite du processus de recherche. Ces propriétés définissent le comportement de toutes les méta-heuristiques pendant la recherche d'une solution, en allant de la plus simple à la plus complexe.

- ✓ Des problèmes d'optimisation avec une fonction objectif et/ou des contraintes qui nécessitent beaucoup de temps de calcul. Certains problèmes d'optimisation réels sont caractérisés par un temps de calcul énorme de la fonction objectif.
- ✓ Des modèles non analytiques de problèmes d'optimisation qui ne peuvent pas être résolus de façon exhaustive : beaucoup de problèmes pratiques sont définis par une boîte noire noyée dans une fonction objectif [42].

2.2.2 Gestion des contraintes :

Beaucoup de problèmes d'optimisation sont liés à des contraintes dures qu'on ne peut pas réduire dans la fonction objectif. Pour permettre aux méta heuristiques de gérer efficacement ces contraintes, différentes stratégies ont été mises au point [50], [51], [52],[8]:

- ✓ Rejet ;
- ✓ Pénalité ;
- ✓ Réparation ;
- ✓ Décodage ;
- ✓ Préservation.

2.2.2.1 Stratégie de rejet :

La stratégie de rejet est très simple : seules les solutions faisables sont conservées durant la recherche et les solutions infaisables sont automatiquement jetées. Cette stratégie est concevable si la portion des solutions infaisables de l'espace de recherche est réduite.

2.2.2.2 Stratégie de pénalité :

Avec la stratégie de pénalité, les solutions infaisables sont prises en compte pendant le processus de recherche. La fonction objectif non contrainte est étendue par une fonction de pénalité qui se chargera de pénaliser les solutions infaisables. C'est l'approche la plus populaire. Il existe beaucoup d'alternatives pour définir les pénalités [53].

2.2.2.3 Stratégie de réparation :

Les stratégies de réparation mettent en œuvre des algorithmes heuristiques (procédure de réparation) qui transforment une solution infaisable en solution faisable [54]. Ces stratégies spécifiques au problème sont employées lorsqu'un opérateur de recherche peut générer des solutions infaisables.

2.2.2.4 Stratégie de décodage :

Cette stratégie utilise des encodages indirects qui transforment la topologie de l'espace de recherche. Lorsqu'on utilise une représentation indirecte, l'encodage n'est pas une solution complète du problème. Un décodeur est nécessaire pour exprimer la solution donnée par l'encodage. Selon l'information présente dans l'encodage, le décodeur aura plus ou moins de travail à faire pour dériver une solution complète [55]. Le décodeur peut être non déterministe.

2.2.2.5 Stratégie de préservation :

Une représentation spécifique et des opérateurs garantissent la génération de solutions faisables. Cette stratégie incorpore des connaissances spécifiques du problème dans la représentation et dans les opérateurs de recherche pour générer uniquement des solutions faisables. Cette stratégie est adaptée à des problèmes spécifiques et ne peut pas être généralisée à tous les problèmes d'optimisation sous contraintes.

2.3 Quelques définitions utiles :

2.3.1 Population initiale :

Bien que la distribution de la population initiale joue un rôle critique dans l'efficacité et l'efficience des algorithmes méta-heuristiques, cette étape est souvent ignorée dans leur conception [8]. Néanmoins, lorsque l'on envisage d'implémenter un algorithme d'optimisation, une attention particulière à la manière de générer la population initiale doit être apportée [7].
si la population initiale ne couvre pas efficacement l'espace de recherche, l'algorithme d'optimisation peut ne pas être en mesure de localiser les points de solution appropriés, manquant ainsi l'optimum global [56] et convergeant vers un optimum local.

2.3.2 La taille de la population :

La taille de la population est un paramètre extrêmement important dans tout algorithme méta-heuristique basé sur la population. Plus une population est grande, meilleures seront ses solutions et plus elle aura besoin de ressources informatiques. Un compromis doit être atteint entre les objectifs contradictoires d'avoir une solution de bonne qualité et le faible besoin de calcul [57]. Aucune règle définitive n'existe sur la façon de choisir la taille de la population, bien que certaines règles de conception pratiques aient été extraites en fonction des algorithmes d'utilisation (PSO, GA...) et de la dimension du problème.

2.3.3 Exploitation et exploration :

Un autre facteur important qui peut fortement influencer les performances des algorithmes méta-heuristiques est la capacité à la fois d'explorer (recherche globale) l'espace de recherche à la recherche de régions d'intérêt et d'exploiter (recherche locale) ces régions afin de localiser la solution optimale. Idéalement, un algorithme d'optimisation qui a ces deux caractéristiques entièrement intégrées devrait être conçu. Cependant, l'exploration et l'exploitation sont des caractéristiques quelque peu exclusives.
La puissance de calcul disponible dans la majorité des situations pratiques est assez limitée. Par conséquent, il est nécessaire d'utiliser judicieusement ce pouvoir. Lors de la résolution d'un problème d'optimisation, un certain équilibre entre l'exploration de l'espace de recherche et l'exploitation de ses régions proéminentes doit être établi. Si l'exploration n'a pas été effectuée,

l'optimiseur peut manquer une ou plusieurs régions proéminentes de l'espace de recherche. Si cette région contient la ou les solutions optimales, l'algorithme ne pourra pas trouver cet optimum ni même une solution dans son voisinage. Ainsi se produira le phénomène indésirable de convergence primature ; l'algorithme convergera vers un optimum local et y sera piégé. La vitesse de convergence de l'algorithme d'optimisation pourrait également être affectée si l'exploration n'a pas été suffisamment promue car les régions proéminentes de l'espace de recherche prendront plus de temps à être, voire jamais, découvertes.

D'autre part, une fois que l'exploration a déterminé une région proéminente où un optimum ou au moins une solution de bonne qualité peut être trouvée, l'algorithme d'optimisation va commencer le processus d'exploration de cette région en concentrant la recherche dans son voisinage immédiat. Si ce processus est interrompu de manière prélature, l'algorithme peut négliger une bonne solution ou même un optimum convergé vers une solution moindre. Par conséquent, et outre le fait que la vitesse de convergence de l'algorithme dépend fortement de ce processus, il est nécessaire de donner au processus d'exploitation le temps nécessaire pour accomplir sa tâche.

Généralement, tous les algorithmes méta-heuristiques ont plusieurs mécanismes pour équilibrer ces deux capacités. un bon équilibre entre les thèmes est nécessaire pour obtenir un optimiseur efficace [58, 59].

2.3.4 Critère d'arrêt :

Selon le problème d'optimisation à résoudre, plusieurs critères peuvent être utilisés pour arrêter la procédure d'optimisation. On pourrait citer [8] :

- **Procédure statique :** dans cette méthode, la fin du processus d'optimisation est connue a priori. Elle peut être mise en œuvre avec un nombre fixe d'itérations, une limitation des ressources de calcul ou un maximum d'évaluations de fonctions objectives. Cette procédure est généralement utilisée lorsqu'une limite est imposée.
- **procédure adaptative :** dans cette méthode, la fin du processus d'optimisation ne peut pas être connue a priori. cela pourrait être mis en œuvre avec un nombre fixe d'itérations non améliorées ou lorsqu'une solution optimale ou satisfaisante est obtenue avec une tolérance d'erreur prédéfinie.

2.3.5 Nombre de particules :

La quantité de particules allouées à la résolution du problème dépend essentiellement de deux paramètres la taille de l'espace de recherche et le rapport entre les capacités de calcul de la machine et le temps maximum de recherche. Il n'y a pas de règle pour déterminer ce paramètre.

2.3.6 Le nombre d'itérations :

Le nombre d'itération influe énormément sur la résolution du problème d'optimisation : un petit nombre d'itérations peut arrêter le processus de recherche prématurément, tandis que trop d'itérations peuvent compliquer les calculs et prendre plus de temps que disponible**.**

2.3.7 Les composantes de la vitesse :

- ✓ Les composantes de la vitesse sont des facteurs très importants pour mettre à jour la vitesse de la particule :

- ✓ Le terme $X \times vi$ est la composante d'inertie qui fournit une mémoire de la direction du vol précédent, il représente une impulsion qui empêche de changer radicalement l'orientation des Particules afin de polariser l'essaim dans le sens du courant.
- ✓ Le terme $C \times e1 \times (\text{persobestpos}\ (i, j) - \text{population}\ (i, j))$ est la composante cognitive qui mesure la performance Des particules par rapport aux performances passées, Elle ressemble à une mémoire Individuelle de la meilleure position pour la particule. L'effet de la composante cognitive Appelée «la nostalgie de la particule » représente la tendance des individus à revenir à des Positions qui les satisfaisaient plus dans le passé.
- ✓ Le terme $C \times e2 \times (\text{globalbestpos}\ (1, j) - \text{population}\ (i, j))$ est la composante sociale qui mesure la performance Des particules i par rapport à un groupe de particules voisines. L'effet de la composante Sociale est que chaque particule vole vers la meilleure position trouvée par le voisinage de la Particule.

2.4 Les méthodes de résolutions des problèmes d'optimisation méta- heuristiques :

2.4.1 Les méthodes de la recherche locale (basée sur une seule solution) :

2.4.1.1 Recuit Simulé (Simulated Annealing, SA) [41] :

Le recuit simulé est une méta-heuristique d'optimisation combinatoire autorisant des variations de cout positives qui permettent d'échapper aux minima locaux dans lesquels le système se piège lorsqu'on utilise des méthodes d'amélioration itératives classiques. C'est une méthode probabiliste de résolution des problèmes d'optimisation combinatoire issue de la physique statique qui s'inspire fortement des techniques du recuit thermique.

Le recuit simulé a obtenu d'excellents résultats sur différents problèmes complexes connus pour leur forte combinatoire : le problème du voyageur de commerce, le problème du recouvrement d'un graphe (Vertex Cover Problem), l'affectation quadratique ou le partitionnement du graph, les problèmes de placement et/ou de routage de circuit électroniques, le car Sequencing Problem, le problème de partitionnement des zones de police et bien d'autres. C'est ce qui a incité un nombre important de chercheurs, toutes disciplines confondues, à s'intéresser à cette méta-heuristique.

Ainsi des propriétés mathématiques, relatives à la convergence vers des solutions optimales globales, ont été établies, Malheureusement, les conditions requises pour cette convergence sont asymptotiques et sont donc, en pratique, irréalisables. Par conséquent, plusieurs schémas pratiques du recuit simulé ont été proposés. Ces schémas sont approximatifs, en ce sens qu'ils conduisent toujours à une solution réalisable mais pas nécessairement à une solution optimale. Les applications citées précédemment, ont montré que cette solution est assez proche et parfois le même que la solution optimale.

a) Recherche Tabou (Tabu Search, TS) [41] :

La recherche tabou est une méta-heuristique qui construit un voisinage en excluant un certain nombre de configurations récemment visitées (ou mouvements récents). Elle utilise toujours la

meilleure solution voisine indépendamment du fait qu'elle améliore ou non la qualité de la solution précédente.

Tant qu'on ne se trouve pas dans un optimum local, cette méthode se comporte comme une méthode de plus forte descente qui améliore à chaque itération la valeur de la fonction objectif en choisissant une configuration meilleure. Mais lorsqu'elle a atteint un optimum local, la recherche tabou cherche à s'en : déplacement dans l'espace de recherche en sélectionnant à chaque itération le meilleur voisin dans un voisinage débarrassé des solutions récentes marquées tabou (pour éviter les cycles). On alterne ainsi entre des phases d'intensification et de diversification qui poussent la recherche à se diriger vers un optimum global. Échapper en choisissant le voisin le moins mauvais, c'est à dire celui qui détériore le moins le coût de la dernière solution sélectionnée. Pour éviter d'emprunter le chemin inverse (retour à une solution précédente) qui provoquerait un cycle, la méthode se déplace d'une configuration à une autre en s'interdisant de revenir sur une configuration récente déjà rencontrée. Elle utilise pour cela une liste taboue qui contient les dernières configurations visitées et interdites pendant un certain nombre d'itérations, le but étant de donner assez de temps à l'algorithme pour lui permettre de sortir du minimum local.

Cependant, stocker dans la liste tabou des configurations entières et vérifier si elles correspondent à la prochaine destination peut être coûteux en espace mémoire et en temps de calcul. Pour éviter ce problème, une alternative courante consiste à stocker et à interdire un ensemble de mouvements récemment utilisés (plutôt que des solutions), ces mouvements pouvant ramener à une configuration déjà visitée.

Dans le cas d'une liste tabou basée mouvements, l'interdiction d'emprunter un chemin ne porte pas uniquement sur les solutions récemment visitées, elle porte plus globalement sur toutes les solutions qui résulteraient de l'un des mouvements présents dans la liste. Le nombre de ces solutions involontairement déclarées tabou dépend de la capacité de stockage de la liste tabou et peut varier très fortement en fonction du problème et de la structure du voisinage. Il arrive même que cette interdiction empêche de visiter des solutions supérieures (à la meilleure configuration déjà rencontrée) n'ayant pas encore été visitées, ce qui rend le processus de recherche plus difficile. Pour ne pas interdire l'accès à ces configurations de qualité, supérieure, on peut ajouter un mécanisme dit d'aspiration qui permet d'enlever le caractère tabou à certains mouvements jugés utiles pour accéder à des solutions de qualité exceptionnelle. D'autres critères d'aspiration plus sophistiqués ont également été proposés.

2.4.2 Les méthodes de recherche évolutive (basée sur une population de solution) :

Les méta-heuristiques à population de solutions, contrairement aux méthodes à solution unique, font évoluer simultanément un ensemble de solutions dans l'espace de recherche. Il y a d'ailleurs souvent une interaction entre les solutions qu'elle soit directe ou indirecte afin de faire évoluer la population. Deux grandes classes de méta-heuristiques à population de solutions sont distinguables : les algorithmes évolutionnaires inspirés de la théorie de l'évolution de Darwin, et les algorithmes d'intelligence en essaim inspirés de biologie ou de l'éthologie. Dans ce paragraphe, nous nous intéressons à cette dernière catégorie en présentant trois méthodes : l'optimisation génétique (GA), l'optimisation par essaims particulaires (PSO), l'optimisation par Colonie d'abeilles artificielles (ABC) [27].

2.4.2.1 Les algorithmes génétiques :

L'algorithme génétique est l'un des algorithmes d'optimisation méta-heuristique les plus célèbres et les plus réussis qui ont eu un grand impact au Sein de la communauté de la recherche. Cet algorithme a été initialement développé par HOLLAND [60] et ses collaborateurs durant les années 1960 et 1970. Depuis lors, l'algorithme a fait l'objet d'études, d'explorations et de développements intensifs.

L'algorithme est un modèle d'évolution biologique basé sur la théorie de la sélection naturelle [61]. À partir d'une population initiale, l'algorithme commencera le processus d'optimisation en générant de nouvelles populations plus adaptées (générations) de chromosomes à l'aide d'opérateurs génétiques tels que le croisement, la recombinaison et la mutation. Les principales étapes de l'algorithme sont données par l'organigramme de la Figure 2.1.

Chaque chromosome $x_h = \{x_{h1}, \ldots, x_{hD}\} \left(h = 1, \ldots, n_{pop}\right)$ est un vecteur de variable de D dimension qui représente une solution possible au problème d'optimisation avec une valeur de fitness $F_{itness}\left(x_h\right)$. Une fois la population initiale générée et la fitness de ses chromosomes évaluée, l'algorithme commence à itérer en suivant les étapes suivantes [62] :

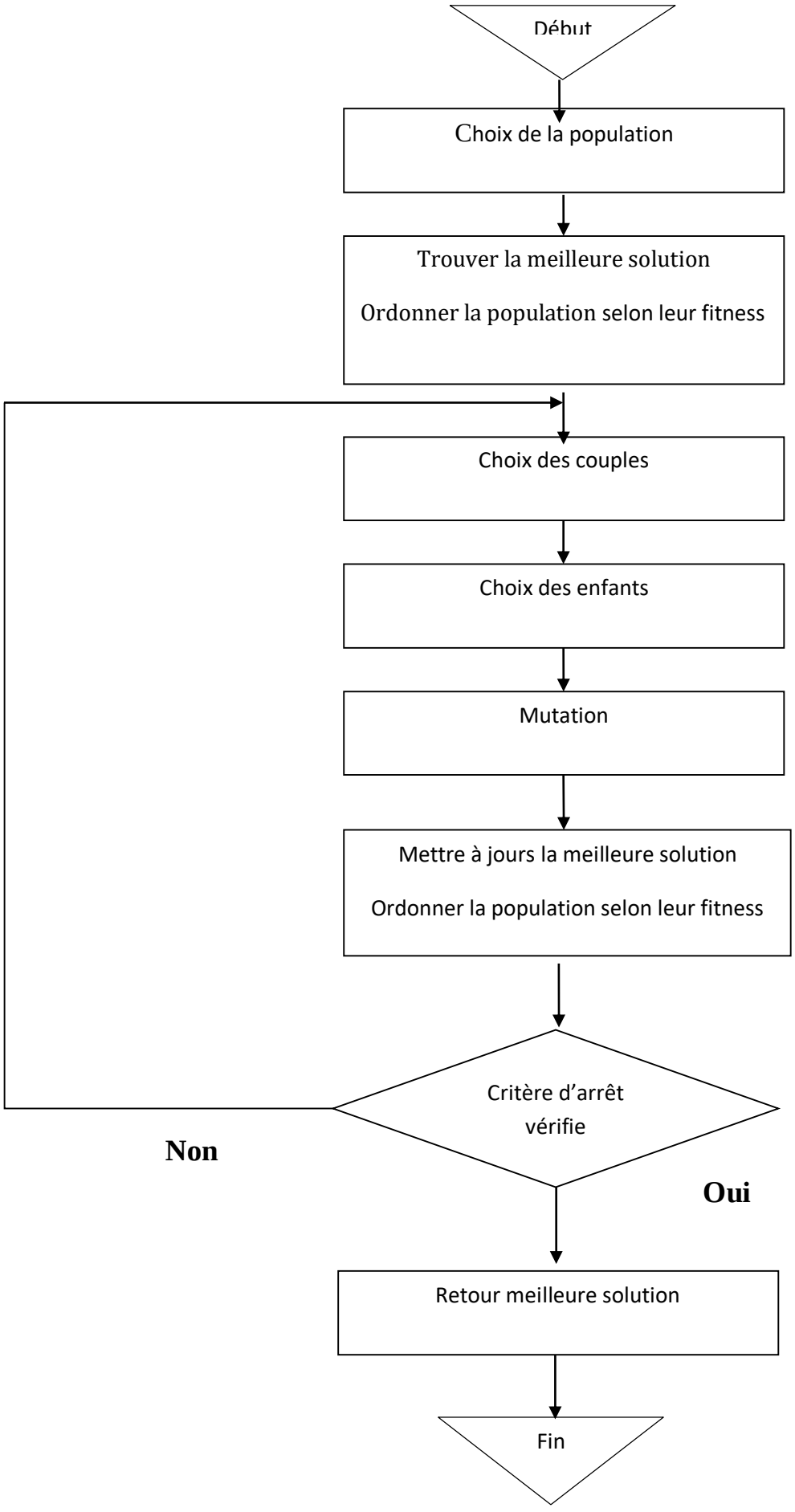


Figure2.1 **:**L'organigramme d'algorithme génétique.

2.4.2.1.1 Les étapes de l'algorithme génétique :

a) Sélection des variables et de la fonction de coût [62] :

Une fonction de coût génère une sortie à partir d'un ensemble de variables d'entrée (un chromosome). La fonction de coût peut être une fonction mathématique, une expérience ou un jeu. L'objet est de modifier la sortie d'une manière souhaitable en trouvant les valeurs appropriées pour les variables d'entrée.

b) La sélection naturelle :

Sur la base des informations de fitness recueillies lors de la dernière phase, les chromosomes seront triés de manière décroissante en fonction de leur fitness. Les chromosomes suffisamment en forme seront sélectionnés pour survivre et éventuellement reproduire la progéniture de la prochaine génération, tandis que les autres mourront. Des chromosomes n_{pop} de la population, seuls les meilleurs membres de $Nkeep$ seront conservés pour l'accouplement. Les chromosomes restants seront retirés pour faire place à la nouvelle progéniture. Ce processus permettra à la population d'évoluer au fil des générations.

c) Sélection :

Il est maintenant temps de jouer aux entremetteurs. Deux chromosomes sont sélectionnés dans le pool d'accouplement des chromosomes $Nkeep$ pour produire deux nouveaux descendants. L'appariement a lieu dans la population d'accouplement jusqu'à ce que la progéniture $n_{pop} - Nkeep$ naisse pour remplacer les chromosomes rejetés.

L'appariement des chromosomes dans un GA peut être aussi intéressant et varié que l'appariement dans une espèce animale. Nous examinerons diverses méthodes de sélection, en commençant par la plus simple.

- **Appairage de haut en bas** : Commencez en haut de la liste et associez les chromosomes deux à la fois jusqu'à ce que les chromosomes $Nkeep$ supérieurs soient sélectionnés pour l'accouplement. Ainsi, l'algorithme associe des lignes impaires à des lignes paires. La mère des numéros de ligne dans la matrice de population donnée par $ma = 1, 3, 5, \cdots$ et le père à les numéros de ligne $pa = 2, 4, 6, \cdots$ Cette approche ne modélise pas bien la nature mais est très simple à programmer. C'est un bon pour les débutants à essayer.
- **Appariement aléatoire** : Cette approche utilise un générateur de nombres aléatoires uniformes pour sélectionner les chromosomes [62].

d) Accouplement (Mating) :

La progéniture sera générée en fusionnant les parents pour transmettre le matériel génétique. La méthode la plus simple consiste à choisir au hasard un ou plusieurs points de croisement dans le chromosome. Le premier descendant sera construit en copiant le premier parent jusqu'au point de croisement, après quoi le deuxième parent sera utilisé. Cette procédure est inversée pour la deuxième progéniture.

Soit $x_f = \{x_{f1}, \ldots, x_{fD}\}$ et $x_m = \{x_{m1}, \ldots, x_{mD}\}$ Soit le parent, puis la progéniture x_1 et x_2 sont donnés :

$$x_1 = \{x_{f1}, x_{f2}, \uparrow x_{m4}, x_{m5}, \ldots, x_{mD}\}$$
$$x_2 = \{x_{m1}, x_{m2}, \uparrow x_{f4}, x_{f5}, \ldots, x_{fD}\}$$

Cette approche consistant à générer des descendants n'est pas attrayante car aucun nouveau matériel génétique n'est introduit une fois qu'une population initiale a été choisie. Nous ne faisons qu'échanger des variables entre les chromosomes ; aucune nouvelle variable ne sera ajoutée aux chromosomes à ce stade.

e) Mutation :

Pour permettre à l'algorithme d'explorer d'autres régions de l'espace de recherche et d'échapper aux optima locaux, un changement ou une mutation dans certaines des variables est introduit de manière aléatoire. Un paramètre appelé taux de mutation est utilisé pour déterminer la probabilité qu'une variable soit mutée. Par exemple, un taux de mutation de 20% indique que 1/5 des variables de tous les chromosomes seront remplacées par des valeurs générées aléatoirement. Les variables à muter sont également choisies aléatoirement.

L'algorithme continuera à itérer en répétant les quatre phases précédentes jusqu'à ce que le critère d'arrêt soit satisfait.

À l'origine, l'algorithme GA utilisait une représentation binaire car les chromosomes étaient représentés par des chaînes binaires de 0 et 1. Cependant, cette représentation discrète ne fonctionnait bien que pour les problèmes nécessitant des solutions de faible dimension et de précision. Pour surmonter cette limitation, le concept de véritable GA codé était introduit [63] où un vecteur de gènes à valeurs réelles a été utilisé pour représenter un chromosome. Les phases restantes de l'algorithme sont les mêmes que dans la représentation binaire.

L'algorithme génétique est l'un des algorithmes d'optimisation les plus largement utilisés dans l'optimisation non linéaire moderne [61], néanmoins, il présente plusieurs lacunes connues [61],[64], à savoir sa tendance à converger vers l'optimum local si la fonction de fitness n'a pas été correctement formulée,

son un taux de convergence lent [63]et l'énorme besoin de calcul nécessaire pour trouver une solution. En fait, étant donné le même problème et le même temps de calcul, des algorithmes d'optimisation plus simples peuvent trouver de meilleures solutions.

Afin de surmonter ces problèmes, l'équilibre entre l'exploration et l'exploitation doit être renforcé. Au sein de l'AG, l'opération de croisement affecte de manière décisive la capacité d'exploration de l'algorithme [63]. En tant que tel, de nombreuses recherches ont été menées sur la manière de produire des opérateurs de croisement plus efficaces [65],[66].

2.4.2.2 Optimisation par essaims particulaires [67] :

L'algorithme d'optimisation par essaims particulaires, Particle Swarm Optimization(PSO), est un algorithme d'optimisation méta-heuristique qui a été introduit par Kennedy et Eberhart en 1995 comme solution au problème d'optimisation d'un seul problème objectif continu. Elle repose sur l'observation du comportement collectif d'insectes sociaux comme les fourmis, les termites et les abeilles ainsi que le comportement d'autres sociétés animales comme les nuées

d'oiseaux ou les bancs de poissons [68]. Fondamentalement, un certain nombre de particules distribuées au hasard commenceront à surfer sur l'espace de recherche à la recherche d'une solution à un problème d'optimisation.
Chaque particule représente une solution potentielle, et au moyen d'une fonction de fitness $(F_{itness}(.))$, l'adéquation de la particule sera évaluée. En utilisant ces données, les particules vont changer leurs positions pour essayer d'améliorer leur aptitude à fournir des solutions plus précises (Figure 2.2).

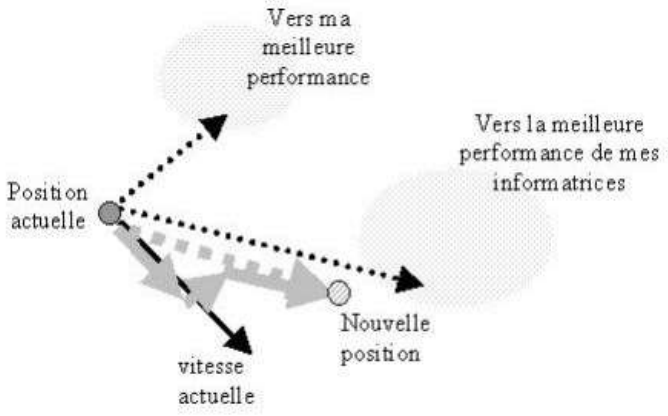


Figure 2.2 : Déplacement d'une particule.

Ce changement est influencé par l'historique du partial ou sa meilleure position, son historique de voisinage, généralement la meilleure position trouvée par l'ensemble de l'essaim, en plus de certains comportements aléatoires.
La première étape consiste à générer la population initiale, à évaluer la forme physique des particules et à déterminer la meilleure solution globale P_g puis l'algorithme commencera à itérer en mettant à jour les positions des particules.
Les mouvements des particules sont liés par les équations suivantes :

$$v(m+1) = v(m) + a(m+1) \tag{2.1}$$

$$X(m+1) = x(m) + v(m+1) \tag{2.2}$$

Où v, a, x et m représentent respectivement la vitesse, l'accélération, la position et l'indice d'itération. Pour l'algorithme proposé par Clerc Kennedy [69], l'accélération d'un i partiel donné, est donnée par l'expression suivante :

$$a_i = \chi\left[c\mathrm{E}_1\left(P_g - X_i\right) + c\mathrm{E}_2\left(p_i - X_i\right)\right] - (1-\chi)v_i \tag{2.3}$$

$\chi = 0,729843788$, $c = 2,05$ et E_1et E_2 sont des nombres aléatoires de la distribution uniforme $U[0,1]$, x_i et $v_i$et la position actuelle et la vitesse de la particule$i$ respectivement, P_i est la meilleure position personnelle de la particule$i$, tandis que P_g est la meilleure position globale de toutes les particules. Notez que a_i peut prendre aussi bien des valeurs négatives que des valeurs positives. Le paramètre χ garantit une vitesse décroissante pour chaque particule

lorsque le nombre d'itérations augmente. Cette propriété améliorera la qualité de la convergence au fur et à mesure que les particules s'approchent de la solution, leur mouvement sera de plus en plus limité, ce qui aidera à affiner la solution trouvée.
L'algorithme PSO a une sérieuse faiblesse qui pourrait sérieusement réduire ses performances si aucune action n'est entreprise. Principalement, l'algorithme PSO a du mal à maintenir un équilibre sain entre l'exploration de l'espace de recherche et l'exploitation de régions importantes de l'espace de recherche [70],[71]. En conséquence directe, l'algorithme d'optimisation serait susceptible d'être piégé dans l'optimum local, en particulier sur des espaces terrestres de fitness multimodaux, robustes et non séparables [72]. Ainsi, l'algorithme sera vulnérable au problème de convergence prématurée.
De nombreuses recherches ont été menées afin de résoudre ce problème et de rendre l'algorithme PSO plus efficace. En effet, diverses variantes de l'algorithme PSO ont été proposées. Par exemple Silva, Neves [71]ont proposé une stratégie prédateur-proie afin de maintenir la diversité. Tandis que He, Wu [73] ont introduit une congrégation passive PSO (PSOPC) dans laquelle les informations peuvent être transférées entre les particules pour éviter de mal juger les informations et d'être piégées par de faibles minima locaux, Sun, Fang [74]ont proposé une optimisation d'essaim de particules à comportement quantique avec point d'attracteur local distribué gaussien. D'autres variantes de l'algorithme PSO peuvent être trouvées dans [75], [76],[77].

2.4.2.3 Optimisation par Colonie d'abeilles artificielles (ABC) :

L'ABC est un algorithme d'optimisation méta-heuristique qui a été introduit par Karaboga en 2005 [22] pour résoudre le problème d'optimisation dans les fonctions multi variables. Il est basé sur l'observation faite sur le comportement social de l'essaim d'abeilles mellifères.
La colonie d'abeilles artificielles contient trois groupes d'abeilles : les abeilles employées, les abeilles spectatrices et les éclaireuses. La première demi-colonie est constituée des abeilles artificielles employées et la seconde comprend les badauds.

- **Abeilles employées :**

Les abeilles employées exploitent une source de nourriture et annoncent sa position aux spectateurs en dansant dans la ruche voisine [67]. Il y a une source d'abeilles employée.

- **Abeilles badauds :**

Les badauds ont tendance à choisir les meilleures sources de nourriture à exploiter davantage en fonction des informations communiquées par les abeilles employées. Par conséquent, la bonne source de nourriture attire plus d'abeilles spectateurs que les mauvaises.

- **Abeilles éclaireuses :**

Lorsque la source de nourriture considérée est épuisée, elle sera abandonnée et ses abeilles employées seront converties en éclaireuse qui choisira au hasard une nouvelle source de nourriture pour remplacer l'ancienne.
Le nombre de sources de nourriture est égal au nombre d'abeilles employées et également égal au nombre d'abeilles spectatrices.

Dans cet algorithme, la position de la source de nourriture représente une solution possible au problème d'optimisation tandis que la quantité de nectar (qualité de la source de nourriture) correspond à la fitness de la solution associée. Cette fitness est généralement évaluée à l'aide

d'une fonction de coût donnée. La taille de la population désignée par PN est la somme du nombre d'abeilles employées (SN) et du nombre d'abeilles spectatrices (LN). Chaque solution $x_h = \{x_{h1}, \ldots x_{hD}\} (h = 1, \ldots, SN)$ est un vecteur de D dimension où D est le nombre de paramètres d'optimisation.

2.4.2.3.1 Principales phases de l'algorithme (ABC) :

Les principales phases de l'algorithme ABC de base (original) sont les suivantes :

a) Phase d'initialisation :

Les abeilles employées sont placées sur des sources de nourriture initiales aléatoires autour de la ruche dans les limites de l'espace de recherche autorisé.

La position de ces sources alimentaires initiales est générée à l'aide de l'expression suivante :

$$x_h = x_{min\,j} + \alpha_{h\,j} \left(x_{max\,j} - x_{min\,j} \right) \tag{2.4}$$

Où : $h = 1, ..., SN$, $j = 1, ..., D$, x_{minj} et x_{maxj} sont respectivement la borne inférieure et la borne supérieure de la dimension j et $\alpha_{hj} \in [0\ 1]$ est uniformément réparti aléatoirement. Chaque source de nourriture x_h est attribuée à une seule abeille employée.

Les sources de nourriture sont soumises à une itération répétée des processus de recherche des abeilles employées, spectatrices et éclaireuses. Le critère de terminaison est choisi pour être soit un nombre maximal d'itérations soit l'obtention d'une tolérance d'erreur spécifique.

b) Phase d'abeille employée (employed bees) :

À cette étape, chaque abeille employée sera générée une nouvelle position de source de nourriture candidate v_h dans le voisinage de son ancienne position de source de nourriture x_h selon l'expression suivante :

$$v_{hj} = x_{hj} + \varnothing_{hj} \left(x_{hj} - x_{kj} \right) \tag{2.5}$$

Où $\{k = 1, ..., SN, k \neq h\}$ et $j \in \{1, \ldots, D\}$ sont des indices uniformément choisis au hasard, $\varnothing_{hj} \in [-1, 1]$ est un nombre aléatoire uniformément distribué.

Si une nouvelle position générée dépasse ses valeurs limites prédéterminées, elle est adaptée pour rester dans la limite de l'espace de recherche. La fitness de la source alimentaire candidate générée v_h est évaluée et comparée à l'aptitude de x_h. Après cela, une sélection gourmande est appliquée pour décider lequel d'entre eux conserver.

Seul un paramètre d'optimisation est mis à jour lors de la génération de la nouvelle source alimentaire candidate.

c) Phase de sélection probabiliste :

Une fois que toutes les abeilles employées ont terminé leur processus de mise à jour, elles partageront la quantité de nectar (fitness) de leurs sources de nourriture dans la ruche avec les spectateurs. En utilisant ces informations, chaque abeille spectatrice choisira au hasard une

source de nourriture à exploiter avec une valeur de probabilité ph. La valeur du ph est donnée par l'expression suivante :

$$P_h = \frac{1/\ f_{itness}\left(x_h\right)}{\sum_{h=1}^{SN} 1/\ f_{itness}\left(x_h\right)} \tag{2.6}$$

Où $f_{itness}\left(x_h\right)$ est la valeur de fitness de la source de nourriture x_h . Clairement, plus la quantité de nectar d'une source donnée est importante, plus sa probabilité d'être choisie par un spectateur est élevée.

d) Phase des abeilles spectatrices (onlookers bees) :

Une fois que toutes les abeilles spectatrices ont choisi leurs sources de nourriture, chacune produira une nouvelle position de source de nourriture candidate v_h au voisinage de la source de nourriture sélectionnée x_h en utilisant l'équation (2.5). Ensuite, la sélection gloutonne entre x_h et v_h est utilisée.

e) Phase d'abeille éclaireuse (scout bees) :

Une source de nourriture x_h est abandonnée après un certain nombre d'essais infructueux pour produire une meilleure source de nourriture dans son voisinage, est épuisée ; ce nombre est désigné par *limit* . La source de nourriture abandonnée par les abeilles employées est remplacée par une nouvelle source de nourriture aléatoire générée par l'abeille éclaireuse. La position de cette source de nourriture est obtenue à l'aide de l'équation (2.4).
La valeur du paramètre $limit$ est donnée par l'expression suivante :

$$limit = SN \times D \tag{2.7}$$

Le paramètre *limit* permet de conserver la diversité au sein de la population ABC en régulant la génération des abeilles éclaireuses. L'emplacement de la meilleure source de nourriture découverte dans tout l'espace de recherche par une abeille artificielle est stocké dans x_{best} sera remplacé par sa position. Les différents de l'algorithme ABC sont donnés par l'organigramme de la Figure 2.3.

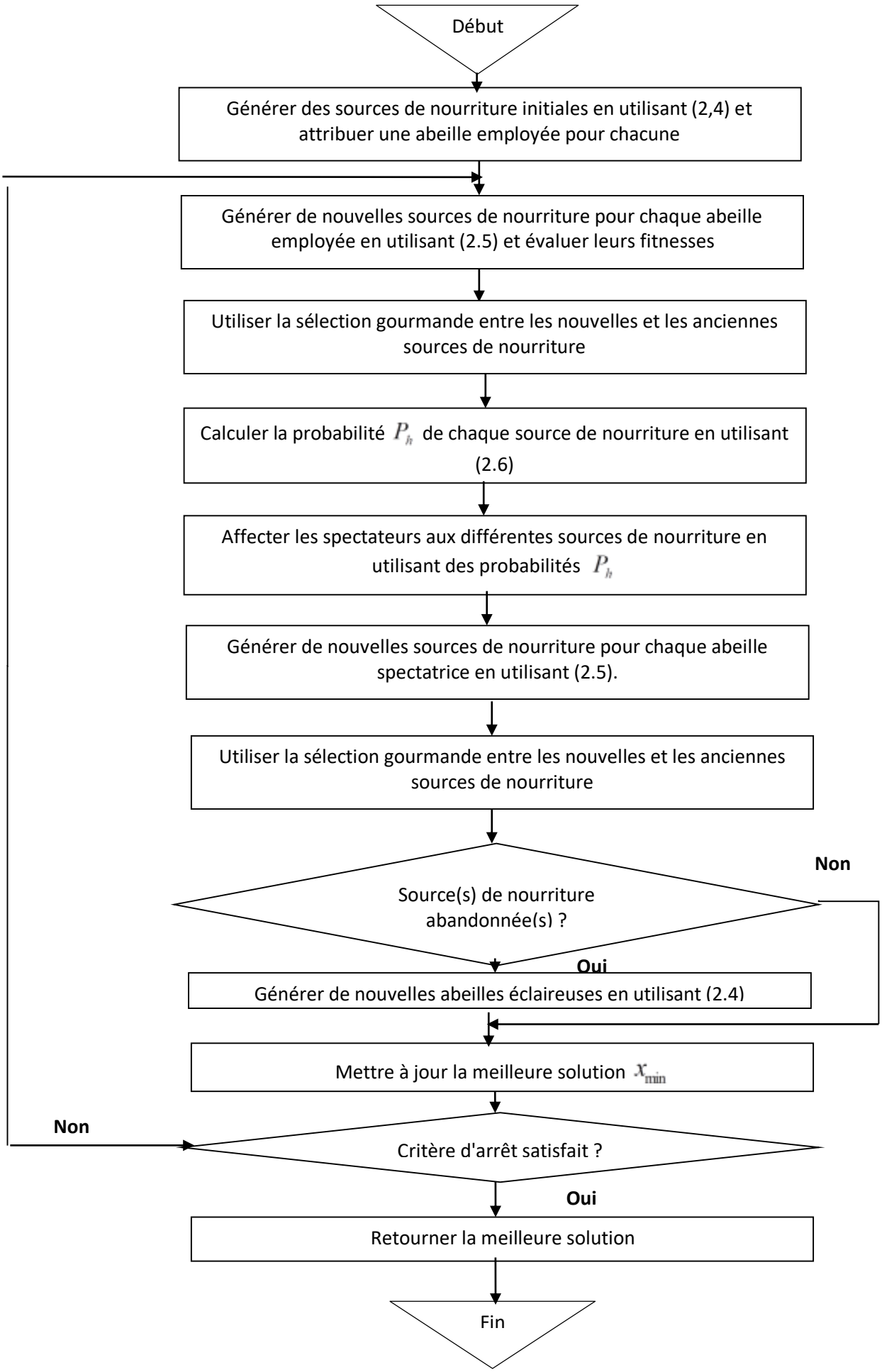


Figure2.3 :l'organigramme d'algorithme ABC.

2.5 Conclusion :

Ce chapitre, on a expliqué en détail les méthodes de résolution des problèmes d'optimisation (présenté dans le chapitre précèdent) par méta-heuristique, leurs caractéristiques et leurs classifications.
En fait, nous sommes intéressés aux algorithmes méta-heuristiques basés sur une population de solutions (i.e, Les méthodes de recherche évolutive) ou on a programmé avec MATLAB code trois algorithmes d'optimisation (i.e, l'algorithme PSO, GA et ABC) afin de faire une comparaison entre les résultats obtenus par ces derniers, cette étude comparative sera l'objectif du prochain chapitre.

Chapitre III :

Étude comparative entre les algorithmes méta-heuristiques

3.1 Introduction :

Dans ce chapitre, on va faire une étude comparative bien détaillée des résultats de trois algorithmes considérés précédemment.

Les résultats ont été obtenus après plusieurs test ou scenarios sur les fonctions de benchmark pour chaque algorithme (GA, PSO et ABC), ou l'objectif de cette étude est de connaitre l'algorithme qui donne les meilleurs résultats et grande efficacité de recherche dans un temps court.

3.2 Étude expérimentale sur les fonctions de référence numériques :

L'approche	Population Initiale	L'équation de mise à jour	Mise à jour de toute la population	Greedy sélection	Mutation
ABC	Aléatoire	$v_{hj} = x_{hj} + \varnothing_{hj}\left(x_{hj} - x_{kj}\right)$	Oui	Oui	oui
PSO	Aléatoire	$v(m+1) = v(m) + \chi\left[c\mathrm{E}_1\left(P_g - X_i\right) + c\mathrm{E}_2\left(p_i - X_i\right)\right] - \left(1-\chi\right)v_i$	Oui	Non	Non
GA	Aléatoire	$x(i,j) = x_{\min} + rand \times (x_{\max} - x_{\min})$	Non	Non	Oui

Tableau 3.1 : tableau comparatif des caractéristiques des algorithmes considérés.

3.3 Fonctions de benchmark :

Les tableaux 3.2 et 3.3 donnent onze fonctions de benchmark [67] fréquemment utilisées pour évaluer les différents algorithmes d'optimisation . Le tableau 3.2 contient six fonctions de benchmark uni-modales tandis que le tableau 3.3 contient cinq fonctions multimodales. Le minimum global de toutes ces fonctions est égal à 0, à l'exception de f_7 qui est égal à -418.9829*D.

Les fonctions	L'espace de recherche	min
$f_1(\vec{x}) = \sum_{i=1}^{D} x_i^2$	[-100,100]	0
$f_2(\vec{x}) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	[-10,10]	0
$f_3(\vec{x}) = max_i \left\{ \lvert x_i \rvert, 1 \le i \le D \right\}$	[-100,100]	0
$f_4(\vec{x}) = \sum_{i=1}^{D-1} 100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2$	[-30,30]	0
$f_5(\vec{x}) = \sum_{i=1}^{D} (x_i + 0.5)^2$	[-100,100]	0
$f_6(\vec{x}) = \sum_{i=1}^{D} i x_i^4 + random(0,1)$	[-1.28, 1.28]	0

Tableau 3.2 : fonctions de référence uni-modèle.

Les fonctions	L'espace de recherche	min
$f_7(\vec{x}) = \sum_{i=1}^{D} x_i \sin(\sqrt{\lvert x_i \rvert})$	[-500,500]	-418.9829*D
$f_8(\vec{x}) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	[-5.2, 5.2]	0
$f_9(\vec{x}) = 20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)$	[-32,32]	0
$f_{10}(\vec{x}) = \frac{1}{4000}\left(\sum_{i=1}^{D}(x_i - 100)^2\right) - \left(\prod_{i=1}^{D}\cos\left(\frac{x_i - 100}{\sqrt{i}}\right)\right) + 1$	[-600,600]	0
$f_{11} = 0.1\left(\begin{array}{l} sin^2(3\pi x_i) + \sum_{i=1}^{D}(x_i - 1)^2(1 + sin^2(3\pi x_i + 1)) + \\ (x_D - 1)^2(1 + sin^2(3\pi x_D)) \end{array}\right)$	-1.28, 1.28]	0

Tableau 3.3 : fonctions de référence multi-modèle.

3.4 Analyse numérique :

Le but de cette analyse est de comparer la qualité de minimisation des algorithmes proposés en considérant l'optimisation numérique des fonctions benchmark des tableaux 3.2 et 3.3.
Pour les onze fonctions, on a considéré plusieurs combinaisons de la taille de la population et le nombre d'itération maximale. Ces scenarios sont indiqués dans le tableau 3.4 .Les résultats de chaque algorithme ont été moyennés sur 50 exécutions afin de prendre en compte la nature aléatoire de ces algorithmes.

ID Scenario	Détails (Taille pop x Iter Max)
1	10 x 100
2	10 x500
3	10 x1000
4	20 x100
5	20 x500
6	20 x1000
7	30 x100
8	30 x500
9	30 x1000
10	50 x100
11	50 x500
12	50 x1000

Tableau 3.4 : combinaisons de la taille de la population et le nombre d'itération maximal.

On a réalisé une étude statistique des résultats obtenus pour le PSO, GA et ABC et cela pour chaque scénario de la taille de la population et le nombre d'itération maximale. Les résultats obtenus ont été compilés dans le tableau 3.5. Nous étions intéressé à calculer la moyenne de la

fonction objective, sa variance, son médiane, la valeur maximale de la fonction objective ainsi que la valeur minimale. Les meilleures valeurs minimales (moyenne, variance, médiane, min et max) pour chaque scenario peuvent indiquer par un fonds gras et un fonds gris.
On peut voir que les résultats obtenus diffèrent énormément pour chaque fonction considérées. Cela est dû aux différentes caractéristiques de chaque cas notamment disposition et forme de la fonction considérée, son espace de recherche et sa nature (uni-model ou multi-model). Une autre remarque qu'on peut citer, et le fait que les trois algorithmes donnent de plus en plus de meilleurs résultats si la taille de la population et le nombre d'itération maximale est de plus en plus incrémenter. Prenant l'exemple de la fonction f_1 entre le premier scénario de (10×100) et le dernier scenario de (50×1000) on peut voir une amélioration dans la solution de l'ordre de 10^{32} pour le PSO, 10^4 pour le GA, et de 10^{15} pour l'ABC.
Le choix de la taille de la population et le nombre d'itération maximal influencent énormément sur la qualité de la solution. Il est clair que l'augmentation de la taille de la population et le nombre d'itération maximal va améliorer la solution, cependant il faut faire attention à la complication numérique de l'algorithme. Au lieu de faire 10×100 =1000 fonction évaluations pour le premier scenario, on aura besoin à faire 50×1000 soit 50 000 fonction évaluations pour le dernier, ce qui signifie un volume de calcul qui est 50 fois plus important avec le dernier scenario, donc, il faut trouver un compromis entre la taille de la population, le nombre d'itérations maximal et la qualité de la solution désirée .

On constate d'après le tableau 2.4 que l'algorithme GA donne les meilleurs résultats. Pour les 132 scenarios considérés, la GA est meilleurs dans 72. Soit un peu plus de la moitie, le PSO est meilleurs dans 31 et l'ABC dans 29 scenarios. l'algorithme GA est meilleur dans les six fonctions : f_3 , f_4 , f_6 , f_8 , f_9 et f_{10} , Tandis que l'algorithme PSO nous a donné les meilleurs solutions en deux fonctions seulement f_1 et f_7 , Alors que l'algorithme ABC a donné la meilleure solution avec fonctions f_2 et f_{11} , il faut noter que, ces solutions ont été obtenues dans huit cas pour la fonction f_2 et neuf cas pour la fonction f_{11} .
Parlant des valeurs minimales et maximales obtenues par ces trois algorithmes et prenant la fonction f_8 comme exemple, parfois on peut trouver une bonne solution mais l'écart entre le min et le max est très grand, cette situation peut être observée avec l'algorithme GA avec le scenario (30×1000) ou la solution moyenne été 5.21E-01, tandis que la solution minimale localisée été égale à 1.51E-09 et la solution maximale égale à 7.97E+00, soit un écart 10E+09 peut marquer.
On peut voir aussi que l'algorithme GA est plus efficace lorsque le nombre d'itération maximale est réduis. Tandis que le PSO donne des meilleurs résultats lorsqu'on augmente la taille de la population et le nombre d'itération maximale.

	NB POP	Iter max	Moyenne			Variance			Médiane			Min			Max		
			PSO	GA	ABC	PSO	GA	ABC	PSO	GA	ABC	PSO	GA	ABC	PSO	GA	ABC
f_1	10	100	1,59E+03	**3,24E+01**	4,57E+02	3,76E+03	**4,71E+01**	3,39E+02	4,30E+02	**1,44E+01**	3,60E+02	4,77E+01	**8,44E-02**	3,34E+01	2,01E+04	**2,17E+02**	2,03E+03
	10	500	6,00E+02	2,81E+00	**1,28E-02**	2,40E+03	4,43E+00	**8,78E-02**	1,06E-03	1,35E+00	**1,50E-05**	6,58E-06	7,45E-03	**4,34E-07**	1,00E+04	1,96E+01	**6,21E-01**
	10	1000	4,00E+02	1,02E+00	**3,81E-11**	1,98E+03	1,76E+00	**2,65E-10**	1,13E-10	3,07E-01	**5,70E-15**	1,61E-15	2,70E-03	**6,66E-17**	1,00E+04	8,34E+00	**1,87E-09**
	20	100	4,54E+02	**1,83E+01**	3,79E+02	1,97E+03	**3,30E+01**	1,66E+02	4,34E+01	**7,04E+00**	3,62E+02	7,16E+00	**2,08E-08**	1,25E+02	1,00E+04	**2,02E+02**	7,71E+02
	20	500	**5,66E-08**	9,90E-01	2,90E-05	**2,91E-07**	2,09E+00	2,85E-05	**3,77E-09**	3,76E-01	1,97E-05	**4,00E-11**	2,56E-05	8,79E-07	**2,06E-06**	1,38E+01	1,25E-04
	20	1000	2,00E+02	4,70E-01	**3,58E-14**	1,41E+03	8,77E-01	**7,56E-14**	**4,53E-22**	1,85E-01	9,13E-15	**4,40E-24**	2,58E-06	2,83E-16	1,00E+04	5,02E+00	**3,77E-13**
	30	100	2,21E+01	**5,88E+00**	3,81E+02	1,65E+01	**1,09E+01**	1,38E+02	1,70E+01	**1,13E+00**	3,66E+02	2,24E+00	**1,61E-03**	1,13E+02	8,19E+01	**5,09E+01**	8,35E+02
	30	500	**1,32E-10**	1,97E-01	4,39E-05	**2,31E-10**	3,17E-01	4,25E-05	**5,34E-11**	6,83E-02	3,22E-05	**3,20E-12**	2,62E-06	1,07E-05	**1,32E-09**	1,91E+00	2,07E-04
	30	1000	**7,56E-25**	5,80E-02	1,77E-13	**2,02E-24**	8,15E-02	1,84E-13	**6,68E-26**	2,65E-02	1,02E-13	**4,27E-28**	1,01E-04	7,58E-15	**9,26E-24**	3,73E-01	7,64E-13
	50	100	6,59E+00	**3,05E+00**	3,75E+02	**4,71E+00**	7,52E+00	1,25E+02	5,30E+00	**2,51E-01**	3,48E+02	1,70E+00	**3,52E-07**	1,63E+02	**2,58E+01**	3,22E+01	8,38E+02
	50	500	**6,07E-13**	6,62E-02	9,84E-05	**1,35E-12**	1,95E-01	5,76E-05	**2,63E-13**	5,64E-03	8,51E-05	**3,09E-15**	1,13E-07	1,72E-05	**8,99E-12**	1,23E+00	2,68E-04
	50	1000	**7,86E-29**	9,87E-03	5,11E-13	**2,69E-28**	2,15E-02	4,65E-13	**5,22E-30**	2,29E-03	3,79E-13	**1,85E-31**	2,99E-07	7,14E-14	**1,80E-27**	1,35E-01	2,67E-12
f_2	10	100	2,89E+01	**8,53E-01**	5,50E+00	2,89E+01	**7,40E-01**	3,44E+00	2,59E+01	**6,54E-01**	4,33E+00	4,24E+00	**6,23E-03**	1,60E+00	6,08E+01	**3,27E+00**	1,81E+01
	10	500	1,60E+01	2,09E-01	**1,87E-05**	1,60E+01	1,78E-01	**2,10E-05**	1,06E+01	1,58E-01	**1,05E-05**	1,03E-01	1,05E-02	**5,48E-07**	4,00E+01	6,98E-01	**8,31E-05**
	10	1000	1,15E+01	1,24E-01	**3,19E-12**	1,15E+01	1,17E-01	**6,80E-12**	1,00E+01	8,02E-02	**4,76E-13**	4,97E-05	3,97E-04	**1,93E-14**	3,00E+01	5,43E-01	**2,87E-11**
	20	100	1,60E+01	**7,17E-01**	7,41E+00	1,60E+01	**6,36E-01**	3,65E+00	1,61E+01	**5,53E-01**	6,31E+00	1,39E+00	**2,06E-02**	2,61E+00	4,31E+01	**3,11E+00**	2,18E+01
	20	500	7,21E+00	1,33E-01	**1,76E-05**	7,21E+00	1,39E-01	**1,28E-05**	1,00E+01	7,57E-02	**1,52E-05**	1,12E-05	4,58E-03	**4,47E-06**	2,00E+01	6,12E-01	**7,31E-05**
	20	1000	7,60E+00	7,82E-02	**1,33E-12**	7,60E+00	8,36E-02	**1,02E-12**	1,00E+01	4,55E-02	**1,13E-12**	6,76E-11	2,31E-03	**1,79E-13**	3,00E+01	3,99E-01	**4,32E-12**
	30	100	1,10E+01	**5,24E-01**	6,67E+00	1,10E+01	**7,01E-01**	2,63E+00	1,11E+01	**2,89E-01**	6,16E+00	8,38E-01	**9,18E-05**	2,90E+00	3,22E+01	**3,32E+00**	1,60E+01
	30	500	5,80E+00	6,49E-02	**2,75E-05**	5,80E+00	6,51E-02	**1,79E-05**	1,13E-03	4,96E-02	**2,35E-05**	**2,07E-07**	1,15E-03	5,48E-06	3,00E+01	2,81E-01	**9,62E-05**
	30	1000	6,40E+00	4,47E-02	**5,05E-12**	6,40E+00	4,55E-02	**3,62E-12**	5,00E+00	2,99E-02	**4,31E-12**	**6,85E-15**	6,24E-05	6,25E-13	2,00E+01	2,09E-01	**1,80E-11**
	50	100	7,26E+00	**2,67E-01**	7,04E+00	7,26E+00	**2,76E-01**	1,76E+00	1,98E+00	**1,46E-01**	6,84E+00	4,40E-01	**1,01E-04**	3,43E+00	3,01E+01	**1,02E+00**	1,23E+01
	50	500	4,20E+00	3,40E-02	**3,63E-05**	4,20E+00	5,25E-02	**1,35E-05**	**6,10E-07**	1,54E-02	3,37E-05	**1,23E-08**	5,25E-04	1,58E-05	2,00E+01	2,14E-01	**8,12E-05**
	50	1000	4,00E+00	2,03E-02	**8,81E-12**	4,00E+00	2,76E-02	**4,42E-12**	**9,44E-16**	1,19E-02	8,03E-12	**1,81E-17**	1,96E-04	2,89E-12	2,00E+01	1,24E-01	**2,32E-11**
f_3	10	100	3,15E+01	**3,02E+00**	5,53E+01	7,79E+00	**2,29E+00**	8,87E+00	3,04E+01	**2,80E+00**	5,57E+01	1,34E+01	**2,53E-02**	3,62E+01	4,91E+01	**9,47E+00**	7,32E+01
	10	500	1,03E+01	**6,70E-01**	3,12E+01	4,80E+00	**4,76E-01**	7,93E+00	9,48E+00	**5,95E-01**	3,22E+01	2,16E+00	**3,44E-03**	1,20E+01	2,29E+01	**2,05E+00**	4,86E+01
	10	1000	2,82E+00	**5,39E-01**	2,60E+01	1,96E+00	**3,60E-01**	7,73E+00	2,47E+00	**4,94E-01**	2,68E+01	3,88E-01	**3,27E-03**	6,71E+00	7,94E+00	**1,57E+00**	3,88E+01
	20	100	1,83E+01	**1,78E+00**	5,55E+01	5,63E+00	**1,68E+00**	7,24E+00	1,79E+01	**1,21E+00**	5,57E+01	7,51E+00	**2,33E-02**	3,99E+01	3,19E+01	**7,57E+00**	7,14E+01
	20	500	1,53E+00	**3,64E-01**	2,31E+01	9,09E-01	**3,47E-01**	5,00E+00	1,33E+00	**2,55E-01**	2,31E+01	1,47E-01	**1,05E-02**	9,65E+00	4,10E+00	**1,51E+00**	3,15E+01
	20	1000	**6,07E-02**	2,49E-01	1,63E+01	**6,94E-02**	2,63E-01	5,73E+00	**3,99E-02**	1,46E-01	1,71E+01	4,51E-03	**2,71E-03**	3,61E+00	**3,67E-01**	1,17E+00	2,67E+01
	30	100	1,50E+01	**1,30E+00**	5,28E+01	4,87E+00	**1,40E+00**	7,76E+00	1,47E+01	**6,87E-01**	5,30E+01	7,23E+00	**4,03E-02**	3,28E+01	2,84E+01	**5,75E+00**	7,57E+01
	30	500	3,59E-01	**2,40E-01**	2,05E+01	**1,97E-01**	2,54E-01	3,71E+00	3,47E-01	**1,25E-01**	1,96E+01	7,43E-02	**1,11E-03**	1,39E+01	**8,38E-01**	1,04E+00	2,84E+01
	30	1000	**4,25E-03**	1,24E-01	1,14E+01	**7,05E-03**	1,29E-01	4,05E+00	**2,65E-03**	8,80E-02	1,10E+01	1,20E-04	**5,46E-05**	4,61E+00	**4,12E-02**	5,74E-01	1,91E+01
	50	100	1,11E+01	**7,43E-01**	5,19E+01	3,49E+00	**8,06E-01**	5,57E+00	1,05E+01	**4,04E-01**	5,13E+01	4,96E+00	**1,39E-02**	4,18E+01	1,78E+01	**3,00E+00**	6,44E+01
	50	500	**6,55E-02**	1,18E-01	1,98E+01	**6,05E-02**	1,04E-01	2,97E+00	**4,13E-02**	1,01E-01	1,97E+01	1,15E-02	**4,46E-03**	1,34E+01	**3,15E-01**	5,13E-01	2,49E+01
	50	1000	**8,67E-05**	5,89E-02	7,06E+00	**8,11E-05**	5,41E-02	1,96E+00	**4,97E-05**	4,45E-02	6,78E+00	**3,70E-06**	7,23E-04	2,63E+00	**3,15E-04**	2,77E-01	1,18E+01

f_4	10	100	3,27E+06	**6,50E+02**	7,82E+05	1,58E+07	**1,62E+03**	7,72E+05	4,22E+04	**5,81E+01**	5,86E+05	1,33E+03	**1,80E+00**	2,33E+04	8,01E+07	**9,15E+03**	3,64E+06
	10	500	1,60E+06	**1,29E+01**	8,92E+02	1,13E+07	**1,24E+01**	1,74E+03	7,23E+01	**1,03E+01**	2,90E+02	1,08E+01	**2,89E-02**	6,31E+00	7,99E+07	**5,66E+01**	8,96E+03
	10	1000	9,47E+03	**6,55E+00**	1,41E+02	2,71E+04	**5,41E+00**	2,64E+02	3,34E+01	**5,05E+00**	7,50E+01	4,77E-01	**4,48E-03**	1,02E+00	9,00E+04	**2,14E+01**	1,40E+03
	20	100	1,90E+04	**3,13E+02**	5,97E+05	3,08E+04	**9,96E+02**	4,01E+05	5,86E+03	**2,38E+01**	5,18E+05	9,06E+02	**5,73E-02**	9,55E+04	1,06E+05	**5,67E+03**	1,64E+06
	20	500	7,40E+03	**8,96E+00**	4,12E+02	2,46E+04	**1,06E+01**	3,96E+02	1,85E+01	**8,07E+00**	2,95E+02	4,12E+00	**2,87E-05**	4,23E+01	9,00E+04	**5,74E+01**	1,95E+03
	20	1000	1,46E+04	**2,34E+00**	1,62E+02	3,33E+04	**3,29E+00**	2,22E+02	1,43E+01	**6,68E-01**	9,87E+01	6,42E-01	**1,11E-04**	1,09E+01	9,00E+04	**1,09E+01**	1,35E+03
	30	100	1,23E+04	**3,06E+02**	4,87E+05	2,83E+04	**1,71E+03**	3,42E+05	1,59E+03	**1,44E+01**	4,25E+05	1,87E+02	**7,33E-03**	5,97E+04	9,11E+04	**1,19E+04**	1,75E+06
	30	500	1,10E+04	**3,38E+00**	3,29E+02	2,95E+04	**3,84E+00**	1,93E+02	1,62E+01	**1,49E+00**	2,55E+02	2,38E+00	**5,05E-02**	3,67E+01	9,00E+04	**1,42E+01**	9,48E+02
	30	1000	7,66E+03	**1,84E+00**	1,22E+02	2,46E+04	**2,88E+00**	1,34E+02	1,33E+01	**5,23E-01**	6,73E+01	6,00E-01	**3,07E-04**	1,52E+01	9,00E+04	**1,12E+01**	6,34E+02
	50	100	8,11E+03	**2,98E+01**	4,68E+05	2,45E+04	**9,12E+01**	2,82E+05	5,93E+02	**9,76E+00**	4,21E+05	9,89E+01	**3,32E-01**	1,08E+05	9,06E+04	**5,89E+02**	1,54E+06
	50	500	5,71E+03	**2,12E+00**	2,71E+02	2,15E+04	**3,48E+00**	1,34E+02	3,01E+01	**2,45E-01**	2,42E+02	1,42E+00	**1,59E-06**	6,04E+01	9,00E+04	**1,46E+01**	6,36E+02
	50	1000	3,87E+03	**1,28E+00**	5,91E+01	1,78E+04	**2,07E+00**	6,00E+01	1,18E+01	**4,38E-01**	2,66E+01	2,19E-02	**1,11E-04**	1,66E+01	9,00E+04	**8,93E+00**	2,86E+02
f_5	10	100	1,84E+03	**3,89E+01**	4,92E+02	3,46E+03	**5,52E+01**	3,86E+02	4,43E+02	**2,05E+01**	3,67E+02	2,52E+01	**1,41E-02**	7,48E+01	1,07E+04	**2,83E+02**	2,00E+03
	10	500	4,00E+02	4,42E+00	**2,13E-04**	1,98E+03	7,75E+00	**6,52E-04**	7,91E-04	1,92E+00	**1,95E-05**	3,19E-06	1,87E-03	**8,45E-07**	1,01E+04	4,85E+01	**4,08E-03**
	10	1000	7,96E+02	1,02E+00	**2,20E-13**	2,73E+03	2,63E+00	**6,76E-13**	2,57E-10	2,61E-01	**4,91E-15**	2,27E-15	3,81E-05	**9,18E-17**	1,01E+04	1,45E+01	**3,80E-12**
	20	100	5,01E+01	**1,32E+01**	4,12E+02	3,59E+01	**1,84E+01**	1,63E+02	3,87E+01	**4,69E+00**	3,96E+02	8,23E+00	**3,44E-03**	1,43E+02	1,64E+02	**1,06E+02**	9,90E+02
	20	500	1,98E+02	9,75E-01	**2,28E-05**	1,40E+03	1,89E+00	**2,10E-05**	**3,66E-09**	3,19E-01	1,69E-05	**4,88E-11**	1,66E-04	3,29E-06	9,90E+03	1,08E+01	**1,20E-04**
	20	1000	4,00E+02	2,14E-01	**3,14E-14**	1,98E+03	3,64E-01	**4,18E-14**	**1,21E-21**	5,14E-02	1,36E-14	**2,10E-24**	2,16E-04	5,77E-16	1,01E+04	1,75E+00	**1,88E-13**
	30	100	2,12E+01	**6,88E+00**	3,60E+02	1,29E+01	**1,25E+01**	1,43E+02	1,76E+01	**1,42E+00**	3,51E+02	5,11E+00	**1,51E-03**	1,01E+02	6,35E+01	**5,42E+01**	7,95E+02
	30	500	**7,25E-11**	3,64E-01	5,38E-05	**1,11E-10**	8,91E-01	5,47E-05	**2,60E-11**	7,71E-02	4,35E-05	**6,77E-13**	1,32E-07	7,28E-06	**5,34E-10**	5,06E+00	3,46E-04
	30	1000	**1,04E-24**	6,96E-02	1,50E-13	**4,01E-24**	1,34E-01	2,48E-13	**1,04E-25**	2,03E-02	6,78E-14	**2,98E-28**	3,89E-09	3,18E-15	**2,74E-23**	7,90E-01	1,40E-12
	50	100	6,39E+00	**3,37E+00**	3,45E+02	**4,21E+00**	7,63E+00	1,08E+02	5,74E+00	**4,53E-01**	3,47E+02	1,23E+00	**2,44E-08**	1,28E+02	**2,29E+01**	3,51E+01	6,19E+02
	50	500	**6,33E-13**	5,11E-02	8,56E-05	**9,38E-13**	1,80E-01	4,77E-05	**3,25E-13**	7,09E-03	7,13E-05	**1,22E-14**	5,04E-07	2,56E-05	**6,03E-12**	1,26E+00	2,34E-04
	50	1000	**4,25E-29**	1,02E-02	6,16E-13	**8,48E-29**	1,62E-02	7,09E-13	**1,11E-29**	3,41E-03	3,80E-13	**1,85E-32**	8,47E-10	8,67E-14	**4,41E-28**	7,16E-02	3,50E-12
f_6	10	100	2,70E+00	**4,73E-02**	1,13E+00	4,76E+00	**2,88E-02**	9,12E-01	5,22E-01	**4,25E-02**	8,52E-01	7,31E-02	**1,36E-03**	2,93E-01	2,46E+01	**1,28E-01**	4,02E+00
	10	500	2,14E+00	**1,52E-02**	1,78E-01	4,75E+00	**9,83E-03**	7,34E-02	1,38E-01	**1,19E-02**	1,59E-01	3,26E-02	**1,35E-03**	5,71E-02	1,88E+01	**4,55E-02**	4,10E-01
	10	1000	2,48E+00	**8,64E-03**	1,48E-01	5,39E+00	**6,48E-03**	5,30E-02	6,87E-02	**7,36E-03**	1,54E-01	1,04E-02	**4,68E-04**	5,82E-02	2,69E+01	**3,32E-02**	2,87E-01
	20	100	6,92E-01	**1,75E-02**	4,95E-01	2,20E+00	**1,28E-02**	1,97E-01	1,73E-01	**1,67E-02**	4,75E-01	4,37E-02	**3,03E-03**	1,60E-01	1,35E+01	**7,38E-02**	1,02E+00
	20	500	8,91E-01	**4,61E-03**	9,47E-02	2,57E+00	**2,93E-03**	3,18E-02	3,05E-02	**4,31E-03**	9,41E-02	5,15E-03	**6,51E-04**	4,40E-02	1,35E+01	**1,38E-02**	1,65E-01
	20	1000	1,31E+00	**3,28E-03**	5,57E-02	3,08E+00	**2,98E-03**	1,76E-02	1,97E-02	**2,00E-03**	5,43E-02	5,58E-03	**2,68E-04**	2,34E-02	1,34E+01	**1,13E-02**	1,11E-01
	30	100	8,74E-01	**1,40E-02**	5,14E-01	2,16E+00	**1,03E-02**	1,97E-01	1,22E-01	**1,13E-02**	5,21E-01	3,95E-02	**2,86E-04**	1,47E-01	1,08E+01	**4,85E-02**	1,02E+00
	30	500	6,66E-01	**3,26E-03**	5,78E-02	2,07E+00	**2,86E-03**	1,67E-02	1,95E-02	**2,40E-03**	5,70E-02	6,42E-03	**4,69E-05**	2,02E-02	1,07E+01	**1,59E-02**	9,50E-02
	30	1000	3,86E-01	**1,81E-03**	4,02E-02	9,41E-01	**1,45E-03**	1,13E-02	9,74E-03	**1,41E-03**	3,72E-02	2,43E-03	**2,23E-05**	1,73E-02	2,71E+00	**5,71E-03**	6,62E-02
	50	100	1,71E-01	**8,05E-03**	3,91E-01	5,30E-01	**5,64E-03**	1,50E-01	6,29E-02	**7,50E-03**	3,83E-01	1,57E-02	**1,60E-04**	1,64E-01	2,76E+00	**2,28E-02**	9,71E-01
	50	500	1,72E-01	**1,78E-03**	3,56E-02	6,43E-01	**1,09E-03**	1,28E-02	1,09E-02	**1,70E-03**	3,18E-02	3,68E-03	**1,55E-05**	1,44E-02	2,69E+00	**4,91E-03**	7,48E-02
	50	1000	1,13E-01	**1,07E-03**	2,61E-02	5,31E-01	**7,58E-04**	7,86E-03	5,99E-03	**9,60E-04**	2,50E-02	1,61E-03	**3,65E-05**	9,64E-03	2,69E+00	**2,94E-03**	4,97E-02

f_7	10	100	**-5,30E+03**	-3,84E+03	-3,47E+03	5,69E+02	4,42E+02	**3,33E+02**	**-5,31E+03**	-4,08E+03	-3,47E+03	**-6,52E+03**	-4,19E+03	-4,43E+03	**-3,57E+03**	-2,91E+03	-2,78E+03
	10	500	**-5,64E+03**	-4,16E+03	-4,32E+03	4,77E+02	**6,14E+01**	6,30E+02	**-5,59E+03**	-4,18E+03	-4,19E+03	**-6,54E+03**	-4,19E+03	-5,87E+03	**-4,82E+03**	-3,88E+03	-3,28E+03
	10	1000	**-5,62E+03**	-4,19E+03	-4,52E+03	5,81E+02	**1,21E+01**	4,23E+02	**-5,57E+03**	-4,19E+03	-4,50E+03	**-6,82E+03**	-4,19E+03	-5,53E+03	-3,84E+03	**-4,11E+03**	-3,74E+03
	20	100	**-5,77E+03**	-4,01E+03	-3,35E+03	6,24E+02	**3,36E+02**	3,42E+02	**-5,74E+03**	-4,15E+03	-3,31E+03	**-7,18E+03**	-4,19E+03	-4,49E+03	**-4,48E+03**	-2,96E+03	-2,76E+03
	20	500	**-6,13E+03**	-4,18E+03	-4,12E+03	5,56E+02	**9,88E+00**	3,57E+02	**-6,05E+03**	-4,19E+03	-4,14E+03	**-7,61E+03**	-4,19E+03	-4,87E+03	**-4,85E+03**	-4,14E+03	-3,53E+03
	20	1000	**-6,07E+03**	-4,19E+03	-4,33E+03	5,36E+02	**1,36E+00**	3,07E+02	**-6,07E+03**	-4,19E+03	-4,33E+03	**-7,55E+03**	-4,19E+03	-5,17E+03	**-4,57E+03**	-4,18E+03	-3,70E+03
	30	100	**-6,08E+03**	-4,00E+03	-3,45E+03	5,81E+02	3,63E+02	**2,94E+02**	**-6,13E+03**	-4,17E+03	-3,41E+03	**-7,14E+03**	-4,19E+03	-4,58E+03	**-4,62E+03**	-3,09E+03	-2,90E+03
	30	500	**-6,21E+03**	-4,19E+03	-4,22E+03	5,55E+02	**3,32E+00**	3,03E+02	**-6,24E+03**	-4,19E+03	-4,17E+03	**-7,21E+03**	-4,19E+03	-4,94E+03	**-4,93E+03**	-4,18E+03	-3,47E+03
	30	1000	**-6,32E+03**	-4,19E+03	-4,40E+03	5,13E+02	**3,75E-01**	3,01E+02	**-6,46E+03**	-4,19E+03	-4,35E+03	**-7,45E+03**	-4,19E+03	-5,06E+03	**-4,88E+03**	-4,19E+03	-3,92E+03
	50	100	**-6,41E+03**	-3,98E+03	-3,43E+03	4,65E+02	3,98E+02	**2,51E+02**	**-6,40E+03**	-4,19E+03	-3,40E+03	**-7,41E+03**	-4,19E+03	-4,05E+03	**-5,42E+03**	-3,05E+03	-3,03E+03
	50	500	**-6,58E+03**	-4,19E+03	-4,17E+03	3,94E+02	**1,10E+01**	2,71E+02	**-6,59E+03**	-4,19E+03	-4,17E+03	**-7,55E+03**	-4,19E+03	-5,00E+03	**-5,63E+03**	-4,11E+03	-3,62E+03
	50	1000	**-6,45E+03**	-4,17E+03	-4,46E+03	4,77E+02	**1,20E+02**	2,74E+02	**-6,45E+03**	-4,19E+03	-4,46E+03	**-7,55E+03**	-4,19E+03	-5,15E+03	**-5,04E+03**	-3,34E+03	-3,88E+03
f_8	10	100	1,14E+02	**8,50E+00**	1,54E+02	3,07E+01	**7,01E+00**	2,14E+01	1,15E+02	**9,56E+00**	1,54E+02	4,95E+01	**1,59E-02**	1,04E+02	1,65E+02	**3,55E+01**	2,09E+02
	10	500	8,27E+01	**1,54E+00**	1,15E+02	3,08E+01	**2,41E+00**	1,56E+01	8,26E+01	**4,42E-01**	1,14E+02	1,59E+01	**7,03E-05**	7,95E+01	1,60E+02	**1,10E+01**	1,52E+02
	10	1000	8,17E+01	**4,34E-01**	1,03E+02	2,71E+01	**8,00E-01**	1,56E+01	7,86E+01	**1,08E-01**	1,08E+02	2,59E+01	**1,05E-04**	5,84E+01	1,59E+02	**3,82E+00**	1,36E+02
	20	100	9,00E+01	**5,66E+00**	1,57E+02	2,78E+01	**6,13E+00**	1,32E+01	9,37E+01	**3,29E+00**	1,61E+02	3,47E+01	**1,55E-03**	1,26E+02	1,64E+02	**2,50E+01**	1,76E+02
	20	500	6,47E+01	**1,32E+00**	1,18E+02	2,08E+01	**2,26E+00**	1,15E+01	6,22E+01	**3,22E-01**	1,17E+02	2,79E+01	**2,42E-04**	9,54E+01	1,11E+02	**1,01E+01**	1,44E+02
	20	1000	5,96E+01	**1,34E-01**	1,10E+02	1,94E+01	**3,45E-01**	1,13E+01	6,03E+01	**3,01E-02**	1,11E+02	1,79E+01	**3,44E-06**	8,35E+01	1,11E+02	**1,65E+00**	1,30E+02
	30	100	8,17E+01	**6,12E+00**	1,48E+02	2,37E+01	**6,67E+00**	1,66E+01	7,94E+01	**4,27E+00**	1,49E+02	3,95E+01	**2,75E-06**	1,09E+02	1,67E+02	**3,27E+01**	1,85E+02
	30	500	4,71E+01	**7,86E-01**	1,13E+02	2,22E+01	**2,24E+00**	9,83E+00	4,23E+01	**6,75E-02**	1,14E+02	1,59E+01	**2,73E-05**	8,45E+01	1,18E+02	**9,48E+00**	1,30E+02
	30	1000	5,03E+01	**5,21E-01**	1,03E+02	2,06E+01	**1,43E+00**	9,99E+00	4,83E+01	**6,30E-02**	1,04E+02	1,69E+01	**1,51E-09**	8,02E+01	1,06E+02	**7,97E+00**	1,22E+02
	50	100	6,78E+01	**8,06E+00**	1,47E+02	1,99E+01	**7,78E+00**	1,47E+01	6,56E+01	**9,20E+00**	1,48E+02	2,58E+01	**4,79E-05**	1,12E+02	1,03E+02	**3,45E+01**	1,72E+02
	50	500	4,08E+01	**5,25E+00**	1,08E+02	1,75E+01	**5,68E+00**	9,46E+00	3,98E+01	**8,33E+00**	1,09E+02	8,00E+00	**4,56E-05**	8,28E+01	9,48E+01	**3,08E+01**	1,27E+02
	50	1000	3,90E+01	**3,15E+00**	9,96E+01	1,62E+01	**4,21E+00**	1,03E+01	3,53E+01	**1,43E-02**	9,93E+01	9,95E+00	**9,38E-07**	7,30E+01	9,96E+01	**1,00E+01**	1,20E+02
f_9	10	100	9,89E+00	**3,33E+00**	1,08E+01	4,79E+00	**2,05E+00**	3,79E+00	7,98E+00	**3,52E+00**	1,03E+01	3,64E+00	**7,79E-02**	5,42E+00	2,02E+01	**9,47E+00**	2,05E+01
	10	500	8,49E+00	**1,12E+00**	1,17E+00	6,23E+00	**9,87E-01**	1,36E+00	6,05E+00	**7,94E-01**	1,15E+00	2,67E-02	3,28E-04	**2,22E-04**	2,00E+01	**4,14E+00**	5,69E+00
	10	1000	7,28E+00	**5,41E-01**	1,16E+00	5,19E+00	**6,04E-01**	1,27E+00	5,47E+00	**2,92E-01**	1,16E+00	1,16E+00	4,59E-03	**5,76E-09**	1,99E+01	**2,33E+00**	5,69E+00
	20	100	4,87E+00	**2,84E+00**	1,09E+01	2,70E+00	**1,54E+00**	3,63E+00	4,42E+00	**3,10E+00**	1,05E+01	2,59E+00	**5,76E-02**	4,90E+00	1,70E+01	**6,53E+00**	1,90E+01
	20	500	1,94E+00	5,41E-01	**5,57E-03**	2,36E+00	6,61E-01	**1,21E-02**	1,65E+00	2,78E-01	**2,54E-03**	**2,96E-06**	1,80E-03	4,30E-04	1,58E+01	2,46E+00	**7,48E-02**
	20	1000	1,96E+00	3,44E-01	**8,01E-08**	3,10E+00	4,60E-01	**1,26E-07**	1,42E+00	1,74E-01	**3,38E-08**	**1,14E-12**	1,95E-02	5,97E-09	1,57E+01	2,36E+00	**6,65E-07**
	30	100	3,30E+00	**1,86E+00**	1,10E+01	**8,92E-01**	1,35E+00	3,66E+00	3,10E+00	**1,65E+00**	1,05E+01	1,70E+00	**9,05E-04**	5,39E+00	6,03E+00	**4,46E+00**	1,96E+01
	30	500	7,64E-01	2,66E-01	**1,04E-02**	8,75E-01	3,28E-01	**2,01E-02**	**1,12E-05**	1,21E-01	3,95E-03	**2,35E-07**	2,37E-03	4,19E-04	2,70E+00	1,76E+00	**1,07E-01**
	30	1000	5,41E-01	9,74E-02	**1,84E-07**	7,90E-01	1,32E-01	**3,43E-07**	**5,22E-13**	5,43E-02	7,84E-08	**2,49E-14**	3,82E-03	1,22E-08	2,32E+00	8,06E-01	**1,77E-06**
	50	100	2,41E+00	**1,10E+00**	1,04E+01	**5,71E-01**	1,21E+00	2,05E+00	2,41E+00	**5,40E-01**	1,02E+01	8,09E-01	**1,95E-03**	7,01E+00	3,41E+00	3,63E+00	**1,48E+01**
	50	500	8,00E-02	**5,85E-02**	8,43E-02	3,22E-01	**7,39E-02**	2,77E-01	**2,51E-07**	2,91E-02	4,43E-03	**7,27E-08**	6,79E-04	9,88E-04	1,42E+00	**2,92E-01**	1,79E+00
	50	1000	1,91E-01	2,39E-02	**1,09E-06**	4,87E-01	2,44E-02	**2,24E-06**	**7,11E-15**	1,51E-02	1,84E-07	**7,11E-15**	5,45E-06	4,62E-08	1,84E+00	1,03E-01	**1,31E-05**

f_{10}	10	100	1,30E+00	**7,28E-01**	2,03E+00	6,73E-01	**3,69E-01**	4,00E-01	1,08E+00	**9,30E-01**	1,97E+00	9,10E-01	**2,44E-02**	1,38E+00	3,61E+00	**1,13E+00**	3,17E+00
	10	500	**2,15E-01**	2,53E-01	6,70E-01	5,50E-01	**3,03E-01**	3,19E-01	**5,52E-02**	8,30E-02	8,18E-01	**9,90E-07**	5,33E-05	1,93E-02	2,64E+00	**9,74E-01**	1,02E+00
	10	1000	3,55E-01	**1,16E-01**	4,19E-01	9,06E-01	**1,37E-01**	3,57E-01	**4,54E-02**	7,00E-02	5,05E-01	**5,66E-14**	4,14E-04	7,00E-07	3,52E+00	**5,63E-01**	9,00E-01
	20	100	8,77E-01	**5,98E-01**	1,81E+00	**1,55E-01**	3,60E-01	2,76E-01	9,20E-01	**7,02E-01**	1,77E+00	4,08E-01	**1,81E-03**	1,22E+00	1,08E+00	**1,04E+00**	2,54E+00
	20	500	3,33E-02	**1,02E-01**	5,98E-01	**3,34E-02**	1,65E-01	1,60E-01	**2,22E-02**	3,66E-02	6,21E-01	**6,32E-12**	3,45E-04	1,89E-01	**1,42E-01**	7,96E-01	8,84E-01
	20	1000	**2,45E-02**	3,26E-02	4,91E-01	**2,75E-02**	5,13E-02	2,28E-01	1,85E-02	**8,75E-03**	5,89E-01	0,00E+00	**2,44E-06**	3,36E-05	**1,40E-01**	2,28E-01	7,94E-01
	30	100	7,41E-01	**3,62E-01**	1,74E+00	**1,90E-01**	3,69E-01	2,02E-01	7,70E-01	**2,34E-01**	1,75E+00	2,40E-01	**1,15E-05**	1,45E+00	1,03E+00	**1,02E+00**	2,15E+00
	30	500	**2,36E-02**	4,01E-02	5,58E-01	**2,35E-02**	8,99E-02	1,37E-01	2,22E-02	**1,15E-02**	5,79E-01	**4,12E-13**	1,69E-07	2,58E-01	**1,03E-01**	5,68E-01	8,29E-01
	30	1000	2,20E-02	**6,55E-03**	5,02E-01	1,98E-02	**1,10E-02**	1,11E-01	1,72E-02	**2,49E-03**	5,24E-01	0,00E+00	**7,27E-06**	2,68E-01	7,88E-02	**5,76E-02**	6,89E-01
	50	100	4,54E-01	**1,34E-01**	1,70E+00	2,37E-01	**2,13E-01**	1,78E-01	4,43E-01	**1,69E-02**	1,67E+00	8,00E-02	**5,07E-05**	1,44E+00	**9,93E-01**	1,02E+00	2,23E+00
	50	500	2,63E-02	**6,68E-03**	5,58E-01	2,92E-02	**1,65E-02**	9,96E-02	1,85E-02	**8,97E-04**	5,48E-01	**4,33E-15**	2,28E-08	2,85E-01	1,86E-01	**8,08E-02**	7,25E-01
	50	1000	2,41E-02	**4,96E-04**	3,43E-01	2,42E-02	**1,27E-03**	1,38E-01	1,72E-02	**6,36E-05**	3,67E-01	0,00E+00	**4,77E-10**	3,23E-02	1,00E-01	**7,36E-03**	5,62E-01
f_{11}	10	100	1,66E-01	**8,54E-02**	1,24E-01	**2,39E-02**	1,06E-01	1,22E-01	1,69E-01	**4,96E-02**	8,72E-02	1,06E-01	**1,55E-04**	2,24E-03	**2,21E-01**	4,99E-01	4,57E-01
	10	500	1,66E-01	**8,42E-03**	8,21E-02	2,61E-02	**1,11E-02**	1,31E-01	1,73E-01	**2,72E-03**	3,82E-02	1,06E-01	1,82E-06	**7,20E-11**	2,02E-01	**4,60E-02**	5,78E-01
	10	1000	1,70E-01	**3,52E-03**	1,51E-02	2,24E-02	**3,94E-03**	3,69E-02	1,73E-01	1,74E-03	**1,55E-04**	7,70E-02	1,20E-06	**2,17E-21**	2,02E-01	**1,64E-02**	2,07E-01
	20	100	1,50E-01	4,10E-02	**1,02E-02**	2,60E-02	5,43E-02	**6,03E-03**	1,54E-01	1,70E-02	**8,33E-03**	5,85E-02	**1,21E-04**	3,14E-03	1,93E-01	2,37E-01	**3,19E-02**
	20	500	1,47E-01	2,60E-03	**2,49E-03**	2,26E-02	**3,89E-03**	1,00E-02	1,48E-01	1,25E-03	**5,19E-10**	8,66E-02	9,59E-08	**9,82E-11**	1,93E-01	**2,06E-02**	4,82E-02
	20	1000	1,53E-01	1,09E-03	**1,06E-04**	1,88E-02	1,61E-03	**7,50E-04**	1,49E-01	5,33E-04	**2,97E-19**	1,06E-01	8,43E-08	**1,46E-20**	1,83E-01	7,60E-03	**5,30E-03**
	30	100	1,36E-01	2,16E-02	**8,99E-03**	2,78E-02	3,08E-02	**4,04E-03**	1,41E-01	1,09E-02	**9,28E-03**	5,78E-02	**1,62E-05**	2,42E-03	1,83E-01	1,69E-01	**2,29E-02**
	30	500	1,37E-01	1,26E-03	**7,65E-04**	2,28E-02	**1,82E-03**	5,41E-03	1,39E-01	5,35E-04	**1,33E-09**	7,70E-02	6,03E-09	**3,15E-10**	1,73E-01	**7,82E-03**	3,82E-02
	30	1000	1,43E-01	7,12E-04	**2,87E-18**	2,57E-02	1,19E-03	**5,40E-18**	1,48E-01	2,21E-04	**1,55E-18**	6,74E-02	6,53E-08	**1,86E-19**	1,83E-01	6,17E-03	**3,83E-17**
	50	100	1,26E-01	1,00E-02	**8,95E-03**	2,31E-02	2,45E-02	**3,74E-03**	1,29E-01	**1,27E-03**	7,58E-03	7,70E-02	**2,31E-07**	4,09E-03	1,64E-01	1,41E-01	**2,15E-02**
	50	500	1,14E-01	3,28E-04	**1,83E-09**	2,92E-02	7,72E-04	**8,97E-10**	1,15E-01	7,58E-05	**1,65E-09**	5,77E-02	2,84E-08	**3,26E-10**	1,73E-01	5,07E-03	**4,23E-09**
	50	1000	1,22E-01	1,69E-04	**1,44E-17**	2,43E-02	3,84E-04	**1,34E-17**	1,21E-01	6,99E-05	**1,03E-17**	6,74E-02	6,94E-09	**1,68E-18**	1,73E-01	2,22E-03	**6,61E-17**

Tableau 3.5 : Résultats comparatifs de la qualité de convergence des benchmark fonctions.

Pour voir l'effet de l'augmentation de la complexité du problème d'optimisation sur les algorithmes Meta-heuristiques considérés, on a procéder à tester les fonctions f_5 (uni-modèle) et f_9 (multi-modèle) et cela on variant le nombre de dimension, de chaque fonction. Pour avoir les valeurs D=5, 10, 20,50 et 100. Les résultats obtenus, pour les mêmes douze scenarios déjà considérés, sont compilés dans les tableaux 3.6 et 3.7.
Il est clair qu'une relation inversement proportionnelle existe entre la qualité de la solution et la complexité du problème d'optimisation indépendamment la fonction, l'algorithme d'optimisation ou le scenario considéré.

D'après le tableau 3.6 (cas de fonction f_5) on peut observer que l'algorithme génétique toujours donne les meilleurs résultats lorsque le nombre de dimension D est égal à 50 et 100 .pour ces deux cas de D, l'algorithme génétique est meilleur dans 20 scenarios parmi les 24, alors que le PSO et l'ABC partagent les 4 scenarios restants. Cependant, pour les valeurs de D entre 10 et 20, le PSO semble donner les meilleurs résultats. L'ABC s'avoir plus efficace lorsque le nombre de dimension est réduis (D=5), avec un meilleur résultat dans 11 scenarios sur 12.

D'après le tableau 3.7 (cas de fonction f_9) les mêmes résultats peuvent être observé l'algorithme GA est toujours meilleur lorsque le nombre de dimension D est égal à 50 et 100. Par contre les résultats des algorithmes PSO et ABC se sont inversées par rapport au nombre de dimension D c'est-à-dire le PSO a donné les meilleurs résultats pour D=5 ou l'ABC est meilleur pour D=10 et 20.

f_5	Nb pop	Iter max	D	moyenne			variance			médiane			min			Max		
				PSO	GA	ABC	PSO	GA	ABC	PSO	GA	ABC	PSO	GA	ABC	PSO	GA	ABC
	10	100	5	**3,57E-02**	7,87E+00	7,79E-01	**1,83E-02**	1,44E+01	2,38E+00	3,47E-02	1,91E+00	**7,83E-03**	**1,25E-09**	2,45E-02	6,88E-07	**7,71E-02**	6,41E+01	1,35E+01
			10	**1,43E+00**	4,37E+01	3,22E+00	**1,92E+00**	6,77E+01	9,21E+00	7,07E-01	1,67E+01	**6,49E-01**	6,26E-02	**4,97E-03**	5,46E-02	**1,09E+01**	3,57E+02	5,31E+01
			20	1,84E+03	**3,89E+01**	4,92E+02	3,46E+03	**5,52E+01**	3,86E+02	4,43E+02	**2,05E+01**	3,67E+02	2,52E+01	**1,41E-02**	7,48E+01	1,07E+04	**2,59E+02**	2,00E+03
			50	2,82E+04	**5,41E+02**	2,43E+04	8,28E+03	**6,24E+02**	4,13E+03	2,77E+04	**3,26E+02**	2,44E+04	1,12E+04	**1,40E+01**	1,66E+04	4,99E+04	**2,57E+03**	3,36E+04
			100	1,10E+05	**7,41E+03**	1,38E+05	1,51E+04	**3,34E+03**	1,30E+04	1,11E+05	**7,24E+03**	1,36E+05	7,82E+04	**1,53E+03**	1,02E+05	1,48E+05	**1,48E+04**	1,67E+05
	10	500	5	3,87E-02	2,45E-01	**2,75E-04**	1,72E-02	3,95E-01	**1,85E-03**	4,25E-02	6,77E-02	**2,39E-08**	**1,56E-32**	8,54E-06	1,31E-16	7,71E-02	2,01E+00	**1,31E-02**
			10	**7,73E-15**	3,48E+00	2,77E-08	**4,54E-14**	5,94E+00	1,71E-07	3,81E-16	9,61E-01	**2,44E-16**	9,20E-19	1,53E-04	**4,48E-20**	**3,22E-13**	2,95E+01	1,20E-06
			20	4,00E+02	4,42E+00	**2,13E-04**	1,98E+03	7,75E+00	**6,52E-04**	7,91E-04	1,92E+00	**1,95E-05**	3,19E-06	1,87E-03	**8,45E-07**	1,01E+04	2,57E+03	**4,08E-03**
			50	9,61E+03	**7,70E+01**	4,69E+02	8,16E+03	**1,15E+02**	3,68E+02	1,04E+04	**2,28E+01**	3,69E+02	2,95E+02	**9,83E-02**	7,51E+01	3,04E+04	**5,51E+02**	2,04E+03
			100	5,89E+04	**4,22E+03**	4,34E+04	1,49E+04	**1,82E+03**	6,99E+03	5,88E+04	**4,26E+03**	4,38E+04	2,08E+04	**6,10E+02**	2,69E+04	9,25E+04	8,26E+03	**6,48E+04**
	10	1000	5	3,96E-02	5,44E-02	**6,32E-07**	1,73E-02	7,40E-02	**1,76E-06**	4,25E-02	3,04E-02	**1,88E-10**	**1,35E-32**	2,30E-06	**0,00E+00**	7,71E-02	3,67E-01	**9,26E-06**
			10	**3,02E-33**	1,25E+00	2,27E-18	**7,81E-33**	1,73E+00	1,55E-17	**0,00E+00**	5,36E-01	2,36E-31	**0,00E+00**	3,39E-03	**0,00E+00**	**4,93E-32**	8,83E+00	1,09E-16
			20	7,96E+02	1,02E+00	**2,20E-13**	2,73E+03	2,63E+00	**6,76E-13**	2,57E-10	2,61E-01	**4,91E-15**	2,27E-15	3,81E-05	**9,18E-17**	1,64E+02	1,45E+01	**3,80E-12**
			50	1,03E+04	3,13E+01	**2,52E+00**	7,69E+03	3,90E+01	**2,63E+00**	1,03E+04	1,50E+01	**1,60E+00**	4,02E+00	2,25E-01	**1,82E-01**	3,01E+04	1,67E+02	**1,18E+01**
			100	4,00E+04	**3,05E+03**	1,60E+04	1,52E+04	**1,33E+03**	4,83E+03	4,07E+04	**2,83E+03**	1,54E+04	1,08E+04	**5,48E+02**	8,59E+03	7,84E+04	**5,86E+03**	3,18E+04
	20	100	5	2,08E-02	2,09E+00	**1,28E-02**	**1,71E-02**	3,46E+00	8,81E-02	1,92E-02	2,06E-01	**5,84E-05**	**7,53E-10**	1,85E-05	2,17E-07	**5,79E-02**	1,63E+01	6,23E-01
			10	**1,32E-01**	3,04E+01	3,74E-01	**1,24E-01**	7,00E+01	2,63E-01	**1,01E-01**	5,42E+00	3,12E-01	**2,03E-02**	4,68E-02	5,94E-02	**6,67E-01**	4,19E+02	1,36E+00
			20	5,01E+01	**1,32E+01**	4,12E+02	3,59E+01	**1,84E+01**	1,63E+02	3,87E+01	**4,69E+00**	3,96E+02	8,23E+00	**3,44E-03**	1,43E+02	9,90E+03	**1,06E+02**	9,90E+02
			50	1,38E+04	**1,55E+02**	3,01E+04	6,49E+03	**2,00E+02**	5,17E+03	1,25E+04	**1,11E+02**	2,96E+04	4,59E+03	**1,52E-03**	2,16E+04	3,04E+04	**1,03E+03**	4,35E+04
			100	7,88E+04	**5,48E+03**	1,52E+05	1,20E+04	**2,61E+03**	1,12E+04	8,00E+04	**5,23E+03**	1,53E+05	5,39E+04	**6,97E+01**	1,26E+05	1,03E+05	**1,14E+04**	1,78E+05
	20	500	5	2,18E-02	6,78E-02	**3,57E-07**	1,63E-02	2,29E-01	**1,46E-06**	1,92E-02	3,38E-03	**2,39E-11**	**1,35E-32**	8,47E-09	1,07E-22	5,21E-02	1,46E+00	**8,88E-06**
			10	**1,41E-19**	6,84E-01	6,54E-12	**4,40E-19**	1,22E+00	4,63E-11	**2,62E-20**	1,01E-01	1,95E-18	**3,50E-22**	1,18E-05	1,17E-19	**3,00E-18**	6,32E+00	3,27E-10
			20	1,98E+02	9,75E-01	**2,28E-05**	1,40E+03	1,89E+00	**2,10E-05**	**3,66E-09**	3,19E-01	1,69E-05	**4,88E-11**	1,66E-04	3,29E-06	9,90E+03	1,08E+01	**1,20E-04**
			50	4,91E+03	**1,84E+01**	2,06E+03	6,16E+03	**3,66E+01**	7,22E+02	1,54E+02	**5,69E+00**	1,90E+03	1,92E+00	**1,79E-02**	6,93E+02	2,02E+04	**2,12E+02**	3,77E+03
			100	2,61E+04	**2,80E+03**	7,03E+04	1,21E+04	**1,26E+03**	7,66E+03	2,51E+04	**2,73E+03**	6,88E+04	3,85E+03	**1,09E+02**	5,34E+04	5,98E+04	**5,53E+03**	8,56E+04
	20	1000	5	2,52E-02	3,91E-03	**2,21E-09**	1,82E-02	7,97E-03	**1,46E-08**	2,50E-02	5,19E-04	**4,84E-15**	1,35E-32	3,27E-09	**6,16E-33**	6,17E-02	3,82E-02	**1,03E-07**
			10	0,00E+00	3,55E-01	**1,04E-17**	**0,00E+00**	6,53E-01	4,97E-17	**0,00E+00**	6,81E-02	**0,00E+00**	**0,00E+00**	5,76E-05	**0,00E+00**	**0,00E+00**	3,33E+00	2,99E-16
			20	4,00E+02	2,14E-01	**3,14E-14**	1,98E+03	3,64E-01	**4,18E-14**	**1,21E-21**	5,14E-02	1,36E-14	**2,10E-24**	2,16E-04	5,77E-16	1,01E+04	1,75E+00	**1,88E-13**
			50	3,02E+03	**6,95E+00**	6,84E+01	5,42E+03	**1,05E+01**	3,98E+01	1,07E-01	**3,00E+00**	5,97E+01	4,46E-04	**1,12E-04**	2,70E+01	2,00E+04	**5,15E+01**	1,91E+02
			100	1,84E+04	**1,72E+03**	4,62E+04	1,07E+04	**9,91E+02**	7,12E+03	2,03E+04	**1,66E+03**	4,62E+04	5,66E+02	**2,98E+01**	3,17E+04	6,02E+04	**4,35E+03**	6,29E+04
	30	100	5	1,47E-02	8,18E-01	**1,72E-04**	1,50E-02	1,56E+00	**5,25E-04**	9,62E-03	1,62E-01	**2,09E-05**	**6,74E-11**	1,17E-07	4,02E-07	5,21E-02	7,03E+00	**3,33E-03**
			10	**4,46E-02**	6,41E+00	2,57E-01	**5,01E-02**	1,63E+01	1,62E-01	**2,37E-02**	1,18E+00	2,37E-01	4,40E-03	**3,83E-05**	2,66E-02	**2,38E-01**	8,54E+01	9,57E-01
			20	2,12E+01	**6,88E+00**	3,60E+02	1,29E+01	**1,25E+01**	1,43E+02	1,76E+01	**1,42E+00**	3,51E+02	5,11E+00	**1,51E-03**	1,01E+02	6,35E+01	**5,42E+01**	7,95E+02
			50	6,04E+03	**1,47E+02**	2,94E+04	3,99E+03	**2,34E+02**	4,51E+03	5,34E+03	**5,89E+01**	2,91E+04	2,02E+03	**1,08E-02**	1,90E+04	2,46E+04	**1,12E+03**	4,03E+04
			100	6,39E+04	**4,56E+03**	1,51E+05	1,22E+04	**2,32E+03**	1,03E+04	6,33E+04	**4,44E+03**	1,53E+05	4,01E+04	**7,47E+02**	1,19E+05	9,63E+04	**1,11E+04**	1,73E+05
	30	500	5	1,74E-02	3,29E-02	**1,27E-10**	1,56E-02	7,17E-02	**7,93E-10**	9,62E-03	2,56E-03	**1,25E-15**	1,35E-32	7,91E-10	**0,00E+00**	5,21E-02	3,12E-01	**5,57E-09**
			10	**9,70E-22**	2,75E-01	7,01E-18	**1,37E-21**	4,65E-01	1,29E-17	**3,60E-22**	8,17E-02	1,66E-18	**5,46E-24**	1,71E-05	1,01E-19	**6,72E-21**	2,48E+00	7,08E-17
			20	**7,25E-11**	3,64E-01	5,38E-05	**1,11E-10**	8,91E-01	5,47E-05	**2,60E-11**	7,71E-02	4,35E-05	**6,77E-13**	1,32E-07	7,28E-06	**5,34E-10**	5,06E+00	3,46E-04
			50	1,81E+03	**1,04E+01**	2,75E+03	3,88E+03	**1,48E+01**	6,77E+02	**2,02E+00**	4,93E+00	2,73E+03	7,64E-02	**9,37E-03**	1,41E+03	1,01E+04	**6,46E+01**	4,63E+03
			100	1,73E+04	**2,23E+03**	7,93E+04	1,04E+04	**1,12E+03**	6,66E+03	1,31E+04	**2,26E+03**	7,81E+04	1,92E+03	**1,63E+02**	6,04E+04	5,48E+04	**4,55E+03**	9,63E+04

.

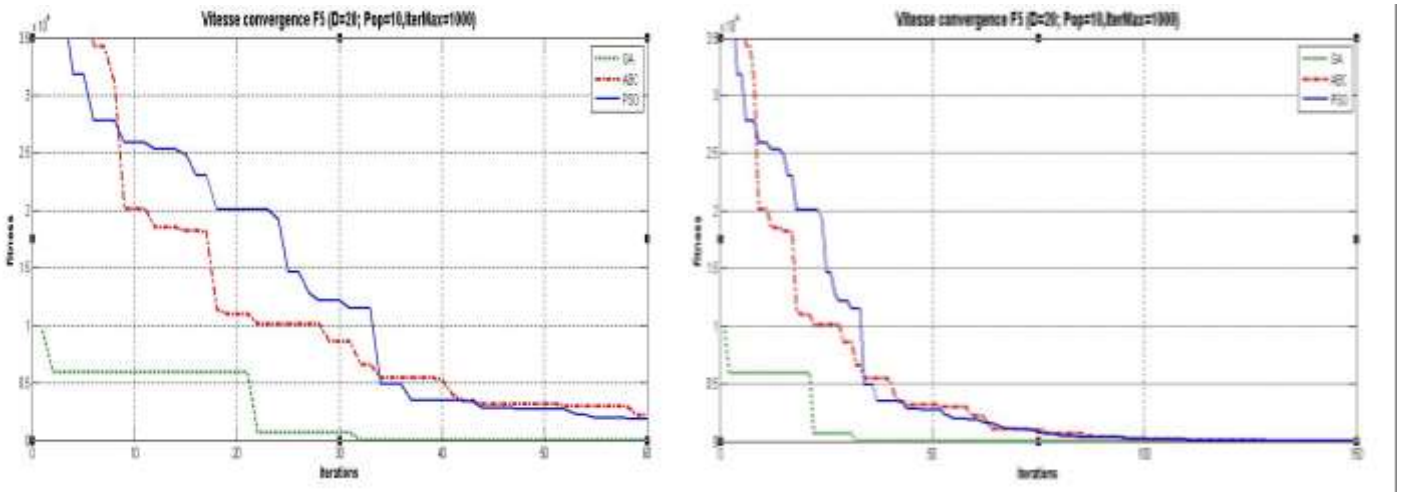
30	1000	5	1,93E-02	1,02E-02	**3,02E-14**	1,51E-02	3,56E-02	**2,06E-13**	1,92E-02	9,24E-04	**1,31E-20**	1,35E-32	9,13E-15	**0,00E+00**	4,25E-02	2,38E-01	**1,46E-12**
		10	**0,00E+00**	5,78E-02	3,13E-22	0,00E+00	8,80E-02	**2,22E-21**	**0,00E+00**	2,74E-02	**0,00E+00**	**0,00E+00**	1,60E-04	**0,00E+00**	**0,00E+00**	4,30E-01	1,57E-20
		20	**1,04E-24**	6,96E-02	1,50E-13	**4,01E-24**	1,34E-01	2,48E-13	**1,04E-25**	2,03E-02	6,78E-14	**2,98E-28**	3,89E-09	3,18E-15	**2,74E-23**	7,90E-01	1,40E-12
		50	9,98E+02	**2,39E+00**	1,59E+02	3,02E+03	**6,62E+00**	6,16E+01	**8,13E-05**	5,79E-01	1,44E+02	**1,32E-06**	1,96E-03	7,22E+01	1,01E+04	**4,53E+01**	4,02E+02
		100	1,22E+04	**5,33E+04**	5,33E+04	9,51E+03	**9,40E+02**	6,32E+03	1,03E+04	**1,54E+03**	5,39E+04	5,80E+01	**1,09E+01**	4,19E+04	3,11E+04	**3,47E+03**	6,51E+04
50	100	5	8,94E-03	2,74E-01	**2,10E-05**	1,17E-02	6,43E-01	**2,90E-05**	9,62E-03	9,48E-03	**8,21E-06**	2,97E-11	**2,49E-11**	1,54E-06	4,25E-02	3,64E+00	**1,27E-04**
		10	**8,85E-03**	3,15E+00	3,02E-01	**6,29E-03**	8,74E+00	2,05E-01	**7,89E-03**	3,34E-01	2,75E-01	7,67E-04	**1,32E-04**	5,06E-02	**2,35E-02**	5,38E+01	1,41E+00
		20	6,39E+00	**3,37E+00**	3,45E+02	**4,21E+00**	7,63E+00	1,08E+02	5,74E+00	**4,53E-01**	3,47E+02	1,23E+00	**2,44E-08**	1,28E+02	**2,29E+01**	3,51E+01	6,19E+02
		50	3,26E+03	**4,75E+01**	2,97E+04	1,98E+03	**8,48E+01**	4,44E+03	2,91E+03	**1,96E+01**	3,03E+04	1,03E+03	**3,55E-02**	1,77E+04	1,15E+04	**4,02E+02**	3,79E+04
		100	5,13E+04	**3,00E+03**	1,57E+05	1,03E+04	**1,74E+03**	8,36E+03	5,08E+04	**2,65E+03**	1,58E+05	3,31E+04	**4,11E+02**	1,39E+05	7,54E+04	**1,03E+04**	1,72E+05
50	500	5	7,06E-03	4,59E-03	**2,81E-17**	1,17E-02	1,20E-02	**1,98E-16**	**1,35E-32**	1,91E-04	1,49E-30	1,35E-32	4,95E-12	**0,00E+00**	4,25E-02	6,99E-02	**1,40E-15**
		10	**2,18E-23**	6,35E-02	1,15E-17	**4,78E-23**	1,15E-01	1,35E-17	**4,04E-24**	8,90E-03	6,76E-18	**9,68E-26**	8,46E-06	6,61E-19	**2,18E-22**	4,54E-01	6,64E-17
		20	**6,33E-13**	5,11E-02	8,56E-05	**9,38E-13**	1,80E-01	4,77E-05	**3,25E-13**	7,09E-03	7,13E-05	**1,22E-14**	5,04E-07	2,56E-05	**6,03E-12**	1,26E+00	2,34E-04
		50	1,60E+03	**4,83E+00**	3,52E+03	3,70E+03	**6,60E+00**	7,44E+02	**5,40E-02**	1,80E+00	3,45E+03	**2,80E-03**	1,70E-02	1,95E+03	1,01E+04	**3,03E+01**	5,48E+03
		100	1,12E+04	**1,60E+03**	8,04E+04	9,11E+03	**7,83E+02**	6,50E+03	1,07E+04	**1,55E+03**	7,99E+04	2,25E+02	**1,78E+02**	6,71E+04	3,16E+04	**3,00E+03**	9,25E+04
50	1000	5	1,60E+03	2,14E-03	**1,65E-25**	1,28E-02	5,34E-03	**9,10E-25**	9,62E-03	**5,57E-05**	0,00E+00	1,35E-32	1,82E-17	**0,00E+00**	4,25E-02	2,59E-02	**6,18E-24**
		10	**0,00E+00**	9,25E-03	**0,00E+00**	**0,00E+00**	1,71E-02	**0,00E+00**	**0,00E+00**	1,87E-03	0,00E+00	**0,00E+00**	1,13E-08	**0,00E+00**	**0,00E+00**	6,97E-02	**0,00E+00**
		20	**4,25E-29**	1,02E-02	6,16E-13	**8,48E-29**	1,62E-02	7,09E-13	**1,11E-29**	3,41E-03	3,80E-13	**1,85E-32**	8,47E-10	8,67E-14	**4,41E-28**	7,16E-02	3,50E-12
		50	7,96E+02	**9,05E-01**	2,38E+02	2,73E+03	**1,18E+00**	5,98E+01	**5,48E-08**	3,68E-01	2,38E+02	**2,55E-09**	3,44E-03	1,16E+02	1,01E+04	**4,93E+00**	3,72E+02
		100	9,83E+03	**1,18E+03**	5,64E+04	8,66E+03	**6,76E+02**	4,62E+03	9,91E+03	**1,09E+03**	5,64E+04	3,17E+00	**2,52E-02**	4,50E+04	3,01E+04	**2,22E+03**	6,61E+04

Tableau 3.6 : Résultats comparatifs de la qualité de convergence de la fonction f_5

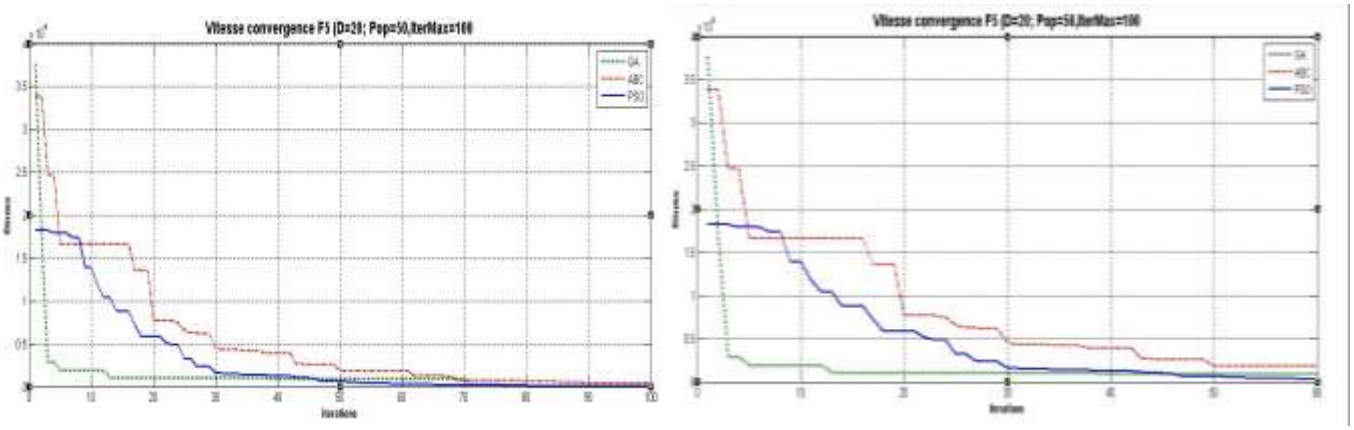
f_9	Nb pop	Iter max	D	moyenne			variance			médiane			min			Max		
				PSO	GA	ABC	PSO	GA	ABC	PSO	GA	ABC	PSO	GA	ABC	PSO	GA	ABC
	10	100	5	**4,86E-01**	1,86E+00	6,37E-01	2,82E+00	1,38E+00	**1,11E+00**	**1,51E-02**	1,81E+00	2,06E-02	2,79E-03	1,28E-02	**4,81E-05**	1,99E+01	5,16E+00	**4,66E+00**
			10	3,30E+00	3,07E+00	**2,07E+00**	4,55E+00	1,82E+00	**1,51E+00**	2,07E+00	3,58E+00	**1,86E+00**	1,85E-01	**1,96E-02**	9,16E-02	1,99E+01	7,62E+00	**5,29E+00**
			20	9,89E+00	**3,33E+00**	1,08E+01	4,79E+00	**2,05E+00**	3,79E+00	7,98E+00	**3,52E+00**	1,03E+01	3,64E+00	**7,79E-02**	5,42E+00	2,02E+01	**9,47E+00**	2,05E+01
			50	1,84E+01	**5,85E+00**	1,98E+01	1,40E+00	2,88E+00	**7,99E-01**	1,87E+01	**5,45E+00**	1,99E+01	1,46E+01	**4,69E-01**	1,80E+01	2,07E+01	**1,20E+01**	2,08E+01
			100	2,01E+01	**1,18E+01**	2,07E+01	3,64E-01	1,57E+00	**1,76E-01**	2,01E+01	**1,20E+01**	2,07E+01	1,93E+01	**7,50E+00**	2,01E+01	2,07E+01	**1,47E+01**	2,09E+01
	10	500	5	3,29E-02	4,21E-01	**1,83E-03**	2,33E-01	5,67E-01	**6,99E-03**	**1,60E-14**	2,19E-01	7,14E-09	3,55E-15	7,52E-05	**2,66E-15**	1,65E+00	2,52E+00	**4,34E-02**
			10	1,77E+00	1,20E+00	**8,06E-01**	3,65E+00	**1,08E+00**	1,11E+00	1,16E+00	8,24E-01	**1,41E-07**	1,37E-09	1,35E-02	**8,89E-11**	1,98E+01	4,20E+00	**3,89E+00**
			20	8,49E+00	1,12E+00	**1,17E+00**	6,23E+00	**9,87E-01**	1,36E+00	6,05E+00	**7,94E-01**	1,15E+00	2,67E-02	3,28E-04	**2,22E-04**	2,00E+01	**4,14E+00**	5,69E+00
			50	**1,73E+01**	3,25E+00	8,65E+00	1,80E+00	**1,10E+00**	2,28E+00	1,76E+01	**3,58E+00**	8,37E+00	1,21E+01	**1,85E-01**	4,68E+00	1,98E+01	**5,23E+00**	1,49E+01
			100	1,92E+01	**9,70E+00**	1,94E+01	**5,32E-01**	2,03E+00	1,01E+00	1,94E+01	**9,90E+00**	1,97E+01	1,75E+01	**3,90E+00**	1,65E+01	2,05E+01	**1,30E+01**	2,08E+01
	10	1000	5	9,88E-02	2,02E-01	**5,43E-05**	3,95E-01	3,33E-01	**3,70E-04**	**3,55E-15**	8,01E-02	5,33E-12	**0,00E+00**	3,50E-03	2,66E-15	1,65E+00	1,61E+00	**2,62E-03**
			10	9,09E-01	4,37E-01	**3,54E-01**	1,07E+00	**4,75E-01**	6,63E-01	5,78E-01	3,31E-01	**7,90E-14**	**3,55E-15**	2,18E-02	6,22E-15	4,17E+00	2,40E+00	**2,01E+00**
			20	7,28E+00	**5,41E-01**	1,16E+00	5,19E+00	**6,04E-01**	1,27E+00	5,47E+00	**2,92E-01**	1,16E+00	1,16E+00	4,59E-03	**5,76E-09**	1,99E+01	**2,33E+00**	5,69E+00
			50	1,66E+01	**2,40E+00**	3,31E+00	2,08E+00	**1,25E+00**	1,35E+00	**1,72E+01**	2,57E+00	3,34E+00	9,16E+00	**3,71E-02**	5,90E-01	1,93E+01	**4,61E+00**	7,33E+00
			100	1,89E+01	**8,63E+00**	1,65E+01	**6,42E-01**	1,62E+00	1,53E+00	1,90E+01	**8,89E+00**	1,64E+01	1,65E+01	**3,85E+00**	1,38E+01	2,02E+01	**1,21E+01**	2,05E+01
	20	100	5	**6,87E-03**	1,58E+00	1,22E-02	**4,43E-03**	1,51E+00	3,19E-02	6,46E-03	1,06E+00	**1,19E-03**	1,23E-03	8,23E-04	**5,58E-05**	**1,97E-02**	5,78E+00	1,73E-01
			10	**5,21E-01**	2,46E+00	1,53E+00	**5,29E-01**	1,80E+00	1,49E+00	**2,70E-01**	2,43E+00	9,04E-01	2,09E-02	**1,43E-03**	9,47E-02	**1,94E+00**	7,32E+00	6,11E+00
			20	4,87E+00	**2,84E+00**	1,09E+01	2,70E+00	**1,54E+00**	3,63E+00	4,42E+00	**3,10E+00**	1,05E+01	2,59E+00	**5,76E-02**	4,90E+00	1,70E+01	**6,53E+00**	1,90E+01
			50	1,58E+01	**4,34E+00**	1,99E+01	2,25E+00	2,21E+00	**5,60E-01**	1,60E+01	**4,22E+00**	2,00E+01	1,13E+01	**1,49E-01**	1,84E+01	2,04E+01	**9,47E+00**	2,07E+01
			100	1,94E+01	**1,05E+01**	2,07E+01	4,40E-01	1,81E+00	**8,24E-02**	1,94E+01	**1,10E+01**	2,07E+01	1,83E+01	**6,56E+00**	2,05E+01	2,03E+01	**1,40E+01**	2,09E+01
	20	500	5	**3,98E-15**	1,43E-01	2,42E-08	**1,85E-15**	2,55E-01	1,67E-07	3,55E-15	4,22E-02	**2,66E-15**	3,55E-15	6,52E-05	**8,88E-16**	**1,42E-14**	1,61E+00	1,18E-06
			10	1,02E-01	7,27E-01	**4,13E-06**	3,55E-01	7,78E-01	**2,92E-05**	**9,15E-11**	3,72E-01	1,32E-10	1,08E-11	2,39E-03	**8,95E-12**	1,65E+00	2,93E+00	**2,07E-04**
			20	1,94E+00	5,41E-01	**5,57E-03**	2,36E+00	6,61E-01	**1,21E-02**	1,65E+00	2,78E-01	**2,54E-03**	**2,96E-06**	1,80E-03	4,30E-04	1,58E+01	2,46E+00	**7,48E-02**
			50	1,14E+01	**1,97E+00**	1,12E+01	4,40E+00	**1,19E+00**	1,96E+00	1,15E+01	**2,01E+00**	1,06E+01	4,00E+00	**2,93E-02**	7,88E+00	2,00E+01	**4,59E+00**	1,62E+01
			100	1,78E+01	**9,13E+00**	2,03E+01	1,02E+00	1,06E+00	**4,04E-01**	1,79E+01	**9,14E+00**	2,05E+01	**1,49E+01**	6,92E+00	1,90E+01	1,91E+01	**1,15E+01**	2,07E+01
	20	1000	5	**1,07E-15**	7,17E-02	3,23E-15	**1,64E-15**	1,54E-01	3,16E-15	**0,00E+00**	2,46E-02	2,66E-15	**0,00E+00**	4,59E-09	8,88E-16	**3,55E-15**	1,04E+00	2,04E-14
			10	5,08E-01	3,85E-01	**3,45E-15**	2,49E+00	4,12E-01	**1,49E-15**	3,55E-15	2,47E-01	**2,66E-15**	**3,55E-15**	9,95E-04	2,66E-15	1,75E+01	1,52E+00	**6,22E-15**
			20	1,96E+00	3,44E-01	**8,01E-08**	3,10E+00	4,60E-01	**1,26E-07**	1,42E+00	1,74E-01	**3,38E-08**	**1,14E-12**	1,95E-02	5,97E-09	1,57E+01	2,36E+00	**6,65E-07**
			50	1,15E+01	**1,24E+00**	4,77E+00	4,02E+00	**1,09E+00**	1,14E+00	1,32E+01	**9,05E-01**	4,45E+00	2,54E+00	**3,09E-03**	2,53E+00	1,82E+01	**4,04E+00**	8,06E+00
			100	1,74E+01	**7,57E+00**	1,89E+01	1,49E+00	1,55E+00	**8,66E-01**	1,76E+01	**7,81E+00**	1,87E+01	1,23E+01	**4,22E+00**	1,71E+01	1,91E+01	**1,01E+01**	2,07E+01

30	100	5	2,96E-03	**9,93E-01**	5,41E-03	**2,30E-03**	1,36E+00	7,93E-03	2,27E-03	2,64E-01	**1,62E-03**	2,88E-04	**6,68E-04**	1,72E-04	**1,17E-02**	5,93E+00	3,80E-02
		10	**3,17E-01**	1,52E+00	1,96E+00	**5,71E-01**	1,53E+00	1,64E+00	**1,21E-01**	1,05E+00	1,65E+00	**2,19E-02**	3,53E-02	9,75E-02	**2,59E+00**	6,77E+00	7,52E+00
		20	3,30E+00	**1,86E+00**	1,10E+01	8,92E-01	**1,35E+00**	3,66E+00	3,10E+00	**1,65E+00**	1,05E+01	1,70E+00	**9,05E-04**	5,39E+00	6,03E+00	4,46E+00	1,96E+01
		50	1,36E+01	**3,65E+00**	1,98E+01	2,47E+00	1,65E+00	**5,04E-01**	1,33E+01	**3,99E+00**	1,98E+01	8,03E+00	**2,28E-01**	1,84E+01	1,90E+01	**8,65E+00**	2,06E+01
		100	1,88E+01	**1,00E+01**	2,07E+01	6,85E-01	2,02E+00	**7,41E-02**	1,90E+01	**1,06E+01**	2,07E+01	1,71E+01	**5,49E+00**	2,06E+01	1,99E+01	**1,32E+01**	2,09E+01
30	500	5	3,27E-15	6,39E-02	**2,24E-15**	**9,74E-16**	1,06E-01	1,17E-15	3,55E-15	1,96E-02	**2,66E-15**	**0,00E+00**	4,25E-05	8,88E-16	3,55E-15	5,18E-01	**2,66E-15**
		10	8,90E-02	2,12E-01	**3,30E-09**	3,60E-01	3,05E-01	**1,49E-08**	**9,49E-12**	1,02E-01	2,50E-10	**6,36E-13**	1,49E-03	1,14E-11	1,65E+00	1,76E+00	**1,04E-07**
		20	7,64E-01	2,66E-01	**1,04E-02**	8,75E-01	3,28E-01	**2,01E-02**	**1,12E-05**	1,21E-01	3,95E-03	**2,35E-07**	2,37E-03	4,19E-04	2,70E+00	1,76E+00	**1,07E-01**
		50	8,32E+00	**1,64E+00**	1,32E+01	4,65E+00	**1,13E+00**	2,44E+00	6,04E+00	**1,61E+00**	1,29E+01	2,64E+00	**7,56E-03**	9,48E+00	1,70E+01	**3,78E+00**	1,85E+01
		100	1,65E+01	**7,78E+00**	2,04E+01	1,67E+00	2,00E+00	**2,86E-01**	1,65E+01	**8,04E+00**	2,04E+01	9,70E+00	**7,29E-01**	1,93E+01	1,89E+01	**1,11E+01**	2,08E+01
30	1000	5	**1,78E-15**	3,12E-02	1,81E-15	1,79E-15	5,63E-02	**1,53E-15**	**1,78E-15**	9,39E-03	2,66E-15	**0,00E+00**	2,06E-05	-8,88E-16	3,55E-15	2,23E-01	**2,66E-15**
		10	2,31E-02	1,49E-01	**2,66E-15**	1,63E-01	2,19E-01	**0,00E+00**	3,55E-15	6,68E-02	**2,66E-15**	3,55E-15	2,10E-04	**2,66E-15**	1,16E+00	1,29E+00	**2,66E-15**
		20	5,41E-01	9,74E-02	**1,84E-07**	7,90E-01	1,32E-01	**3,43E-07**	**5,22E-13**	5,43E-02	7,84E-08	**2,49E-14**	3,82E-03	1,22E-08	2,32E+00	8,06E-01	**1,77E-06**
		50	7,03E+00	**8,40E-01**	5,95E+00	4,28E+00	**8,57E-01**	1,53E+00	5,39E+00	**5,39E-01**	5,57E+00	1,88E+00	**6,11E-03**	4,08E+00	1,55E+01	**3,13E+00**	1,10E+01
		100	1,59E+01	**6,70E+00**	1,95E+01	1,84E+00	1,66E+00	**7,37E-01**	1,63E+01	**7,02E+00**	1,96E+01	1,05E+01	**2,84E+00**	1,80E+01	1,86E+01	**9,62E+00**	2,06E+01
50	100	5	**1,48E-03**	7,70E-01	2,75E-03	**8,07E-04**	1,02E+00	3,76E-03	1,29E-03	2,87E-01	**9,61E-04**	3,68E-04	**4,30E-06**	1,02E-04	**4,60E-03**	3,35E+00	1,70E-02
		10	**4,95E-02**	1,29E+00	1,93E+00	**2,98E-02**	1,30E+00	1,43E+00	**4,34E-02**	9,98E-01	1,83E+00	1,56E-02	**1,26E-02**	1,74E-01	**1,57E-01**	4,39E+00	6,80E+00
		20	2,41E+00	**1,10E+00**	1,04E+01	5,71E-01	**1,21E+00**	2,05E+00	2,41E+00	**5,40E-01**	1,02E+01	8,09E-01	**1,95E-03**	7,01E+00	3,41E+00	**3,63E+00**	1,48E+01
		50	1,06E+01	**2,81E+00**	1,99E+01	2,31E+00	1,79E+00	**3,39E-01**	1,01E+01	**3,23E+00**	1,99E+01	7,15E+00	**7,55E-02**	1,87E+01	1,61E+01	**6,98E+00**	2,04E+01
		100	9,18E+00	2,07E+01	**8,57E-01**	1,66E+00	**5,65E-02**	1,80E+01	**9,36E+00**	2,07E+01	1,55E+01	**6,10E+00**	2,06E+01	1,95E+01	**1,24E+01**	2,08E+01	1,80E+01
50	500	5	2,77E-15	9,39E-02	**6,75E-16**	**1,49E-15**	3,49E-01	1,78E-15	3,55E-15	7,39E-03	**8,88E-16**	**0,00E+00**	2,55E-07	8,88E-16	3,55E-15	2,38E+00	**2,66E-15**
		10	2,31E-02	1,35E-01	**6,39E-07**	1,63E-01	2,14E-01	**4,30E-06**	**9,65E-13**	4,37E-02	2,30E-10	**1,28E-13**	1,57E-04	2,33E-11	1,16E+00	1,02E+00	**3,05E-05**
		20	8,00E-02	**5,85E-02**	8,43E-02	3,22E-01	**7,39E-02**	2,77E-01	**2,51E-07**	2,91E-02	4,43E-03	**7,27E-08**	6,79E-04	9,88E-04	1,42E+00	**2,92E-01**	1,79E+00
		50	4,86E+00	**1,06E+00**	1,38E+01	4,24E+00	**1,07E+00**	2,06E+00	2,84E+00	**6,85E-01**	1,31E+01	1,38E+00	**1,36E-03**	9,79E+00	1,55E+01	**3,60E+00**	1,91E+01
		100	7,05E+00	2,05E+01	**2,21E+00**	1,47E+00	**1,55E-01**	1,45E+01	7,34E+00	2,05E+01	**6,32E+00**	**2,84E+00**	2,01E+01	1,74E+01	**9,71E+00**	2,07E+01	1,41E+01
50	1000	5	7,11E-16	1,26E-02	**3,55E-17**	**1,44E-15**	2,54E-02	1,57E-15	0,00E+00	3,66E-03	**8,88E-16**	**0,00E+00**	3,69E-07	8,88E-16	3,55E-15	1,51E-01	**2,66E-15**
		10	3,55E-15	1,96E-02	**2,66E-15**	7,18E-16	3,29E-02	**0,00E+00**	3,55E-15	9,96E-03	**2,66E-15**	**0,00E+00**	2,63E-04	2,66E-15	7,11E-15	2,00E-01	**2,66E-15**
		20	1,91E-01	2,39E-02	**1,09E-06**	4,87E-01	2,44E-02	**2,24E-06**	**7,11E-15**	1,51E-02	1,84E-07	**7,11E-15**	5,45E-06	4,62E-08	1,84E+00	1,03E-01	**1,31E-05**
		50	4,08E+00	**4,69E-01**	6,81E+00	3,62E+00	**5,90E-01**	1,49E+00	2,70E+00	**2,08E-01**	6,55E+00	**1,41E-04**	3,45E-03	4,62E+00	1,36E+01	**2,57E+00**	1,24E+01
		100	6,26E+00	1,98E+01	**3,28E+00**	1,56E+00	**6,36E-01**	1,39E+01	6,27E+00	2,01E+01	**4,59E+00**	**2,04E-02**	1,82E+01	1,81E+01	**8,96E+00**	2,06E+01	1,31E+01

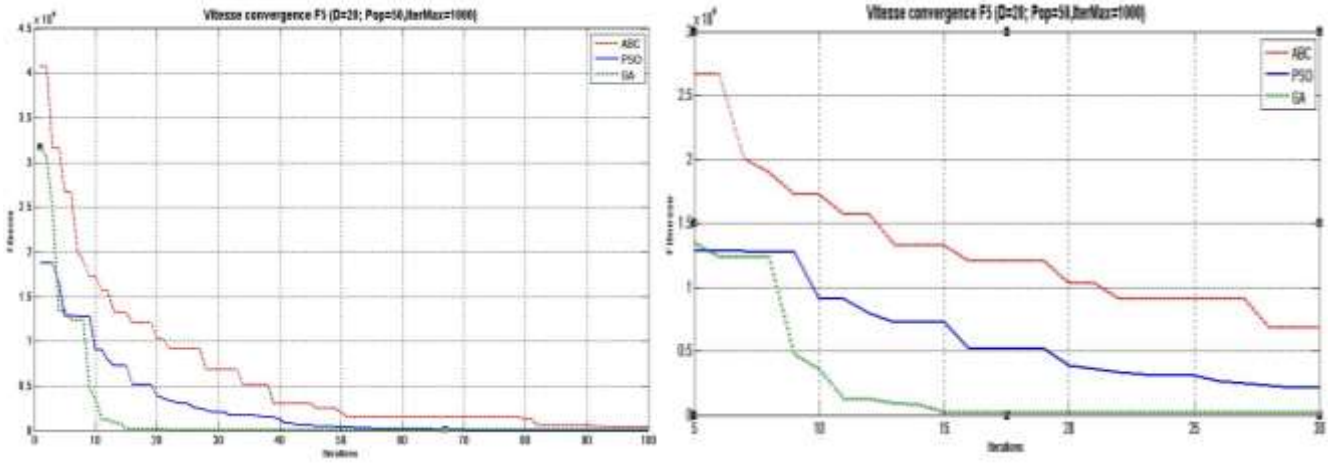
Tableau 3.7 : Résultats comparatifs de la qualité de convergence de la fonction f_9 .



(a)



(b)



(c)

Figure 3.1 : vitesse de convergence de la fonction f_5 ou D=20.

Figure 3.1 repesente la vitesse de convergence de la fonction f_5 ou D=20 en trois scenarios (3, 10 et 12).

Les figures 3.1et 3.2 illustrent la vitesse de convergence des trois algorithmes considérés. Pour le scenario de la fonction f_5 avec D=20 et f_9 avec D=50 successivement d'après la figure 3.1(a).

On peut observer que l'algorithme GA converge plus rapidement vers la solution optimale, la figure (a) est un exemple, ou cet algorithme est plus rapide tout le long de l'intervalle d'itération [0-100].mais son amélioration n'est pas continue ou on peut remarquer que dans l'intervalle [3-21] l'algorithme n'a pas améliorer sa valeur, donc on risque de converger vers un optimum local. De l'autre part les autres algorithmes ABC et PSO ont pratiquement la même vitesse de convergence.

La figure (c) représente le scenario 12 (nombre d'itération maximal (1000) et le nombre de population (10)). Au début l'algorithme PSO semble donner les meilleurs résultats avant que le GA prend sa place à partir d'un nombre d'itération égal à 7 pour continuer au même rythme.

Par contre l'ABC a la vitesse de convergence la plus basse des trois algorithmes.

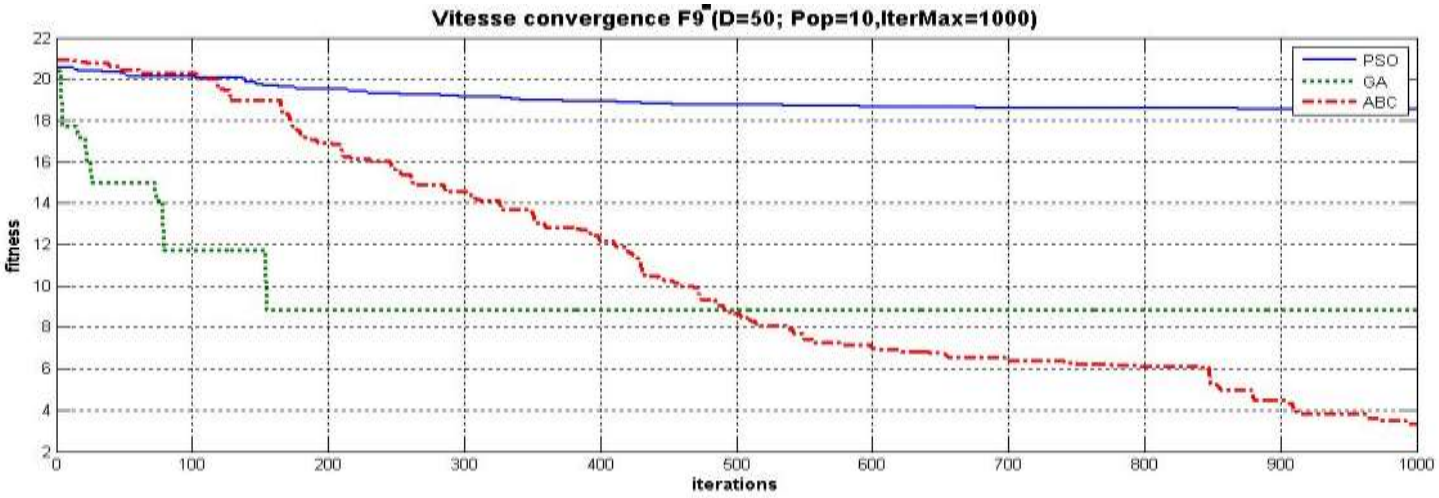
Figure 3.2 repesente la vitesse de convergence de la fonction f_9 ou D=50 en trois scenarios (3, 10 et 12).

Pour les trois cas (a), (b) et(c), on observe un même comportement. Au début le GA a une vitesse plus rapide. Cependant, dès que le nombre d'itération commence à augmenter, l'ABC sera plus rapide pour le cas (b) et le PSO pour le cas (c).

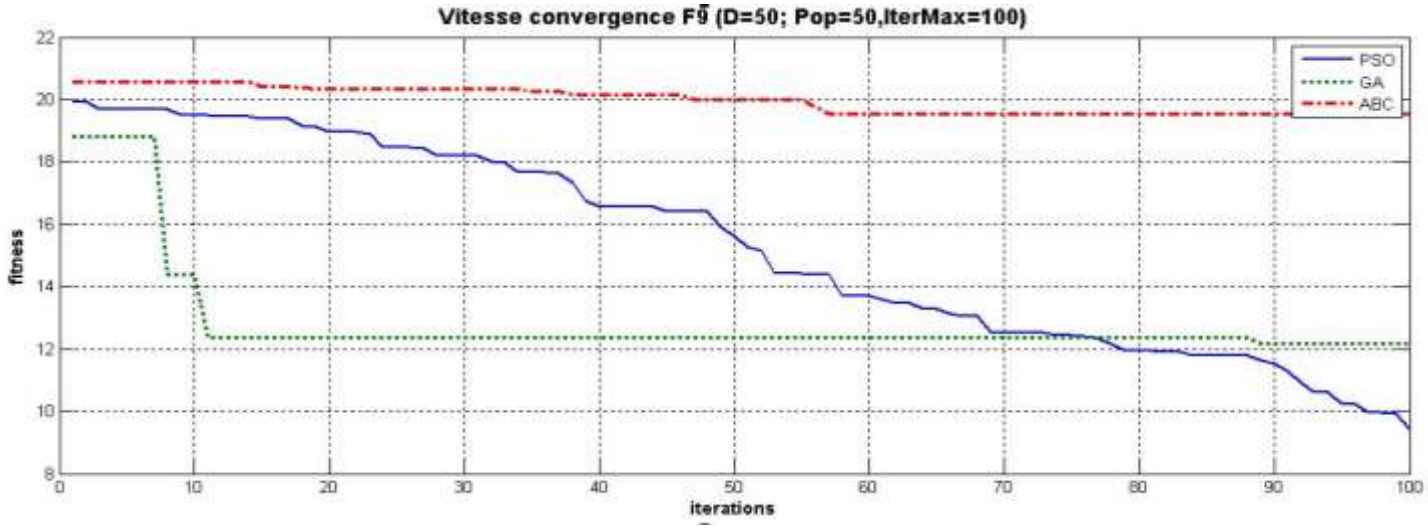
La Figure 3.3 : vitesse de convergence de la fonction f_{10} ou D=5, 10, 20,50et100 de chaque algorithme prenant le scenario 1(10×100).

La figure (a, b et c) indiquent que les trois algorithmes Meta heuristiques considérés décroisent leur vitesse de convergence à chaque fois que le nombre de dimension augmente.

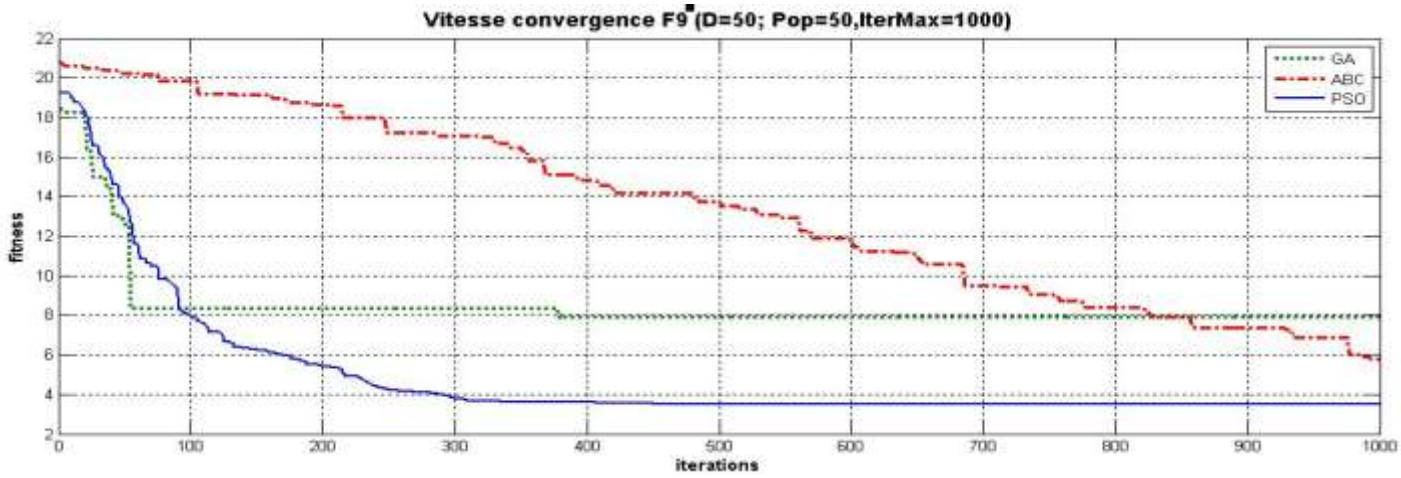
La figure (c) 3.3 est un exemple on peut constater que l'algorithme PSO nous a donner les meilleurs résultats ou le nombre de D=5 suivie par D=10 puis D=20, D=50 enfin D=100 ce qui confirme que l'augmentation du nombre de dimension faire retarder la vitesse de convergence de la fonction objective en fonction de nombre d'itérations des algorithmes d'optimisation méta-heuristique.



(a)

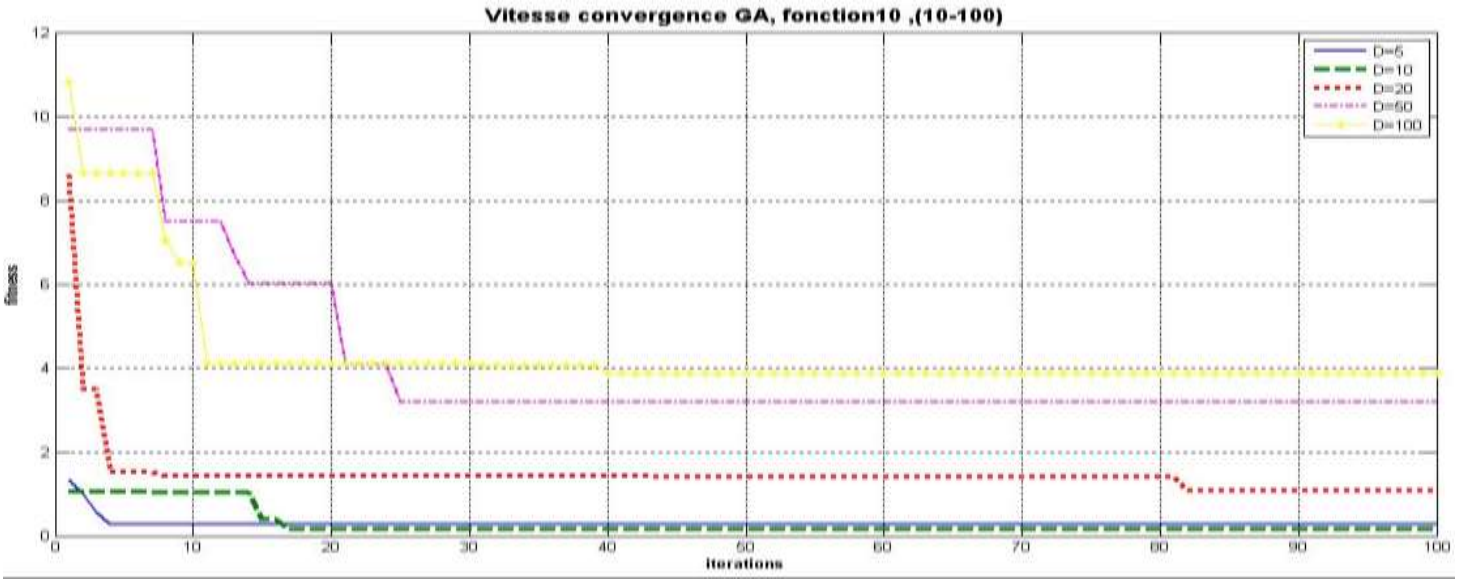


(b)

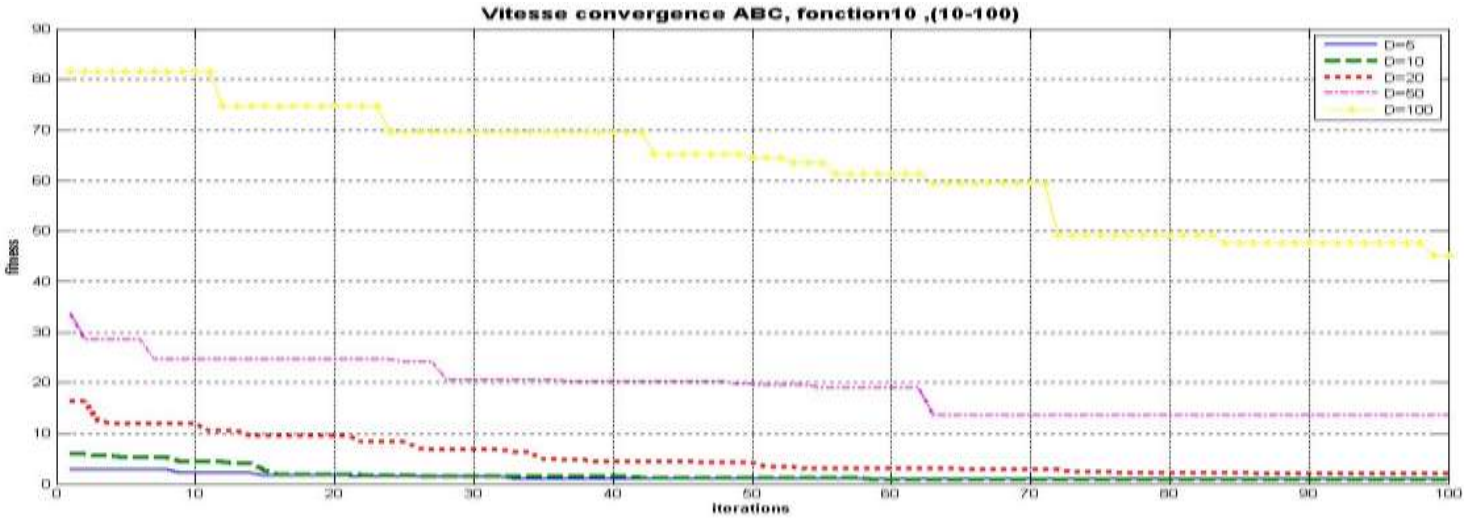


(c)

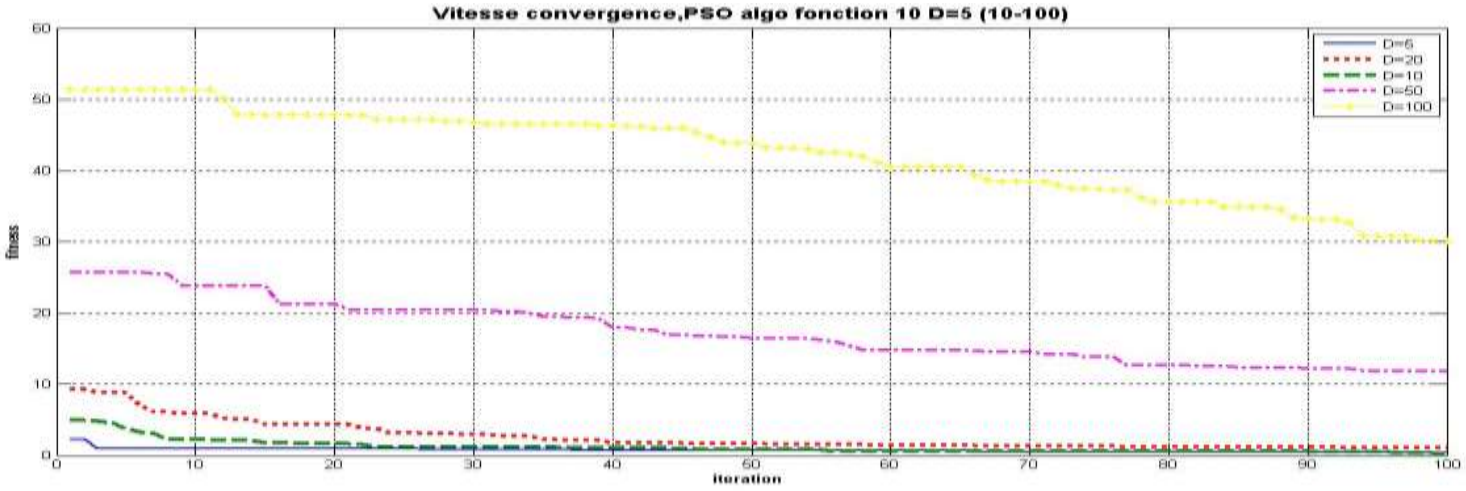
Figure 3.2 : vitesse de convergence de la fonction f_9 ou D=50.



(a)



(b)



(c)

Figure 3.3 : vitesse de convergence de la fonction f_{10} ou D=5, 10, 20,50et100 de chaque algorithme

Nb pop	Iter max	PSO (ms)	GA (ms)	ABC (ms)
10	100	67.65	**48.22**	114.58
10	500	316.07	**211.36**	549.01
10	1000	685.35	**417.98**	1100
20	100	133.19	**75.67**	233.45
20	500	645.12	**381.76**	1140
20	1000	1290	**700.66**	2240
30	100	214.59	**116.07**	352.95
30	500	946.81	**551.33**	1790
30	1000	1920	**1130**	3510
50	100	322.75	**187.56**	633.27
50	500	1700	**904.14**	3210
50	1000	3230	**1770**	6220

Tableau 3.8 : le temps d'exécution de f_5 pour chaque algorithme en m-secondes.

Nb pop	Iter max	PSO (ms)	GA (ms)	ABC (ms)
10	100	67.67	**47.58**	116.26
10	500	376.15	**213.91**	583.83
10	1000	729.69	**427.52**	1113
20	100	130.59	**76.81**	228.81
20	500	657.67	**362.38**	1138
20	1000	1293	**760.59**	2256
30	100	193.80	**116.69**	364.01
30	500	972.45	**570.99**	1746
30	1000	1907	**1154**	3488
50	100	320.99	**192.18**	628.08
50	500	1630	**920.68**	3077
50	1000	3282	**1844**	6168

Tableau 3.9 : le temps d'exécution de f_9 pour chaque algorithme en m-secondes.

Le tableau 3.8 et 3.9 représentent le temps moyen d'exécution (50) de chaque algorithme en millisecondes prenant l'exemple de la fonction f_5 et f_9 avec D=20.

Les résultats de tableau 3.8 et 3.9 montrent que l'algorithme GA est le plus rapide pour les 12 scenarios considères et pour les deux fonctions (multi-modèle et uni-modèle) suivi par la PSO et après l'ABC. Donc quel que soit la complexité du problème l'algorithme GA reste le plus rapide

3.5 Conclusion :

Dans ce chapitre, on a fait une étude comparative pour tester l'efficacité des algorithmes méta-heuristiques (PSO, GA et ABC) utilisant onze fonctions dites fonctions de benchmark avec différents scenarios i .e, la taille de population et nombre d'itérations) en changeant le nombre de dimension, notant la moyenne et la vitesse de convergence avec le temps d'exécution de chaque algorithme après 50 répétitions.

Cette comparaison montre que l'Algorithme Génétique donne les meilleurs résultats lorsque le nombre d'itération maximale est réduit. Alors que, le PSO a une qualité de minimisation importante en cas de nombre d'itérations maximales et une taille de la population élevés.

Conclusion générale

Conclusion générale :

Notre travail s'inscrit dans le domaine de la résolution des problèmes d'optimisation combinatoire difficiles non-linéaires avec l'utilisation des méthodes dites Meta heuristiques pour trouver dans un temps de calcul raisonnable, une solution optimale précise à un problème d'optimisation non linéaire.

Trois algorithmes d'optimisation méta-heuristique ont été testés (PSO, GA et ABC) sur onze fonctions (uni-model et multi-model) dite les fonctions de benchmark, prendre en considération le nombre de population, le nombre d'itérations maximal et le nombre de dimensions au but de faire une comparaison détaillée sur les résultats de leurs moyennes des fitness obtenues (fonction objectif) ainsi que leurs vitesse de convergence et le temps d'exécution de chaque algorithme considéré .

Cette étude montre que l'Algorithme GA est plus efficace et donne toujours les meilleurs résultats de moyenne lorsque le nombre d'itération maximale est réduit. Tandis que, l'algorithme PSO est le meilleur en cas d'augmentation de la taille de la population et le nombre d'itération maximale. Cependant l'algorithme ABC donne de meilleurs résultats lorsque le nombre de dimension est réduit.

On peut conclure aussi que l'algorithme GA commence à converger plus rapidement par rapport aux deux algorithmes restants, mais seulement au début car après un nombre d'itérations sa vitesse se ralentit. D'autre part, on peut observer que l'augmentation du nombre de dimension influe négativement sur le comportement des algorithmes méta-heuristiques.

Les résultats du temps d'exécutions indiquent que le GA est le plus rapide de ces trois algorithmes quel que soit le scenario ou la complexité des fonctions benchmark utilisées.

Finalement, on a choisi ces trois algorithmes pour les tester mais il existe d'autres algorithmes importants ou peut les intégrés dans des problèmes réels, par exemple, une optimisation de la puissance maximale (Photovoltaïque) cas d'ombrage partiel.

Références :

[1] J. L. Cohon, *Multiobjective Programming and Planning. Mathematics in science and engineering.*, 1978.

[2] J.-B. Hiriat-Urruty,Edition1, "Les Mathématiques du mieux faire. vol. 1",1993.

[3] H. Alice Yalaoui, FaroukYalaoui et Lionel Amodeo, "méthodes et techniques " 2012.

[4] A. El Dor, M. Clerc, and P. Siarry, "A multi-swarm PSO using charged particles in a partitioned search space for continuous optimization," *Computational Optimization and Applications,* vol. 53, pp. 271-295, 2012.

[5] E. Matagne, "Cours elec 2311: Physique interne des convertisseurs électromécaniques. milieux magnétiques composites. Notes de cours de l'Université Catholique de Louvains, Belgique," ed.

[6] M. Bierlaire, *Introduction à l'optimisation différentiable*: PPUR presses polytechniques, 2006.

[7] H. Maaranen, K. Miettinen, and A. Penttinen, "On initial populations of a genetic algorithm for continuous optimization problems," *Journal of Global Optimization,* vol. 37, pp. 405-436, 2007.

[8] E.-G. Talbi, *Metaheuristics: from design to implementation* vol. 74: John Wiley & Sons, 2009.

[9] J. C. M. M. Ngirabanzi, "Optimisation et contrôle," 2021.

[10] S. G. Cohen and D. E. Bailey, "What makes teams work: Group effectiveness research from the shop floor to the executive suite," *Journal of management,* vol. 23, pp. 239-290, 1997.

[11] C. Solnon, "Contributions à la résolution pratique de problèmes combinatoires," Université Lyon 1, 2005.

[12] D. Bouyssou, D. Dubois, M. Pirlot, and H. Prade, "Concepts et méthodes pour l'aide à la décision 3: analyse multicritère (Traité IC2, série Informatique et Systèmes d'Information)," ed: Lavoisier, 2006.

[13] M. Abdellaoui and C. Gonzales, "Théorie de l'utilité multiattribut," *Concepts et méthodes pour l'aide à la décision,* vol. 3, pp. 25-62, 2006.

[14] S. B. Mena, "Introduction aux méthodes multicritères d'aide à la décision," *BASE,* 2000.

[15] V. E. Giard and B. Roy, *Méthodologie multicritère d'aide à la décision*: Editions Economica, 1985.

[16] A. Schärlig, *Décider sur plusieurs critères: panorama de l'aide à la décision multicritère* vol. 1: PPUR presses polytechniques, 1985.

[17] C. H. Papadimitriou, "Combinatorial optimization," *Algorithms and complexity,* 1982.

[18] D. H. Wolpert and W. G. Macready, "No free lunch theorems for search," Technical Report SFI-TR-95-02-010, Santa Fe Institute1995.

[19] J. Puchinger and G. R. Raidl, "Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification," in *International work-conference on the interplay between natural and artificial computation*, 2005, pp. 41-53.

[20] G. B. Dantzig, "Maximization of a linear function of variables subject to linear inequalities," *Activity analysis of production and allocation,* vol. 13, pp. 339-347, 1951.

[21] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, 1984, pp. 302-311.
[22] R. E. Gomory, "Outline of an algorithm for integer solutions to linear programs and an algorithm for the mixed integer problem," in *50 Years of Integer Programming 1958-2008*, ed: Springer, 2010, pp. 77-103.
[23] P. C. Gilmore and R. E. Gomory, "A linear programming approach to the cutting-stock problem," *Operations research*, vol. 9, pp. 849-859, 1961.
[24] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance, "Branch-and-price: Column generation for solving huge integer programs," *Operations research*, vol. 46, pp. 316-329, 1998.
[25] E. L. Johnson, "Modeling and strong linear programs for mixed integer programming," in *Algorithms and model formulations in mathematical programming*, ed: Springer, 1989, pp. 1-43.
[26] A. H. Land and A. G. Doig, "An automatic method for solving discrete programming problems," in *50 Years of Integer Programming 1958-2008*, ed: Springer, 2010, pp. 105-132.
[27] A. N. e. P. SIARRY, "Métaheuristiques adaptatives d'optimisation continue basées sur des méthodes d'apprentissage," doctorat, informatique, 2017.
[28] R. E. Bellman and S. E. Dreyfus, "Applied dynamic programming," in *Applied Dynamic Programming*, ed: Princeton university press, 2015.
[29] P. Van Hentenryck and L. Michel, "Constraint-based local search, pages xii–xii," ed: The MIT Press, 2005.
[30] R. Barták, "Constraint programming: In pursuit of the holy grail," in *Proceedings of the Week of Doctoral Students (WDS99)*, 1999, pp. 555-564.
[31] T. Frühwirth and S. Abdennadher, "Market Overview," in *Essentials of Constraint Programming*, ed: Springer, 2003, pp. 101-103.
[32] K. Apt, *Principles of constraint programming*: Cambridge university press, 2003.
[33] F. Rossi, "P. v. BEEK et T. WALSH: Handbook of Constraint Programming (Foundations of Artificial Intelligence)," ed: Elsevier Science Inc., New York, NY, USA, 2006.
[34] A. Letort, "Passage à l'échelle pour les contraintes d'ordonnancement multi-ressources," Ecole des Mines de Nantes, 2013.
[35] J. W. S. Emile Aarts and Jan K. Lenstra, *Local Search in Combinatorial Optimization*. New York, NY, USA, , 1997.
[36] J.-K. Hao, P. Galinier, and M. Habib, "Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes," *Revue d'intelligence artificielle,* vol. 13, pp. 283-324, 1999.
[37] V. Vazirani, "Approximation algorithms springer-verlag," *New York,* 2001.
[38] N. Schabanel., *Algorithmes d'approximation et Algorithmes randomisés*, 2003.
[39] A. A. T.A.J. Nicholson, "Optimization in Industry : Industrial applications.," ed. London Business School series., 1971.
[40] M. Widmer, "Les métaheuristiques: des outils performants pour les problèmes industriels," in *3ème Conférence Francophone de MOdélisation et SIMulation MOSIM*, 2001, pp. 25-27.
[41] B. BELIN, "Conception interactive d'environnement urbains durables à base de résolution de contraintes," Thèse de Doctorat, École doctorale : Sciences et technologies

de l'information, et mathématiques, l'Université de Nantes Angers Le Mans, france, 2014.

[42] C. H. Papdimitriou and K. Steiglitz, "Combinatorial optimization: algorithms and complexity," ed: Prentice Hall, Englewood Cli s, New Jersey, 1982.

[43] P. Festa and M. G. Resende, "An annotated bibliography of GRASP–Part II: Applications," *International Transactions in Operational Research*, vol. 16, pp. 131-172, 2009.

[44] M. R. Gary and D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-completeness," ed: WH Freeman and Company, New York, 1979.

[45] A. Hertz, *LES MÉTA-HEURISTIQUES: quelques conseils pour en faire bon usage*: Groupe d'études et de recherche en analyse des décisions, 2005.

[46] S. Kirkpatrick, "CDG Jr., and MP Vecchi," *Optimization by simulated annealing. Science*, vol. 220, pp. 671-680, 1983.

[47] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & operations research*, vol. 13, pp. 533-549, 1986.

[48] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*: MIT press, 1992.

[49] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM computing surveys (CSUR)*, vol. 35, pp. 268-308, 2003.

[50] Z. Michalewicz, D. Dasgupta, R. G. Le Riche, and M. Schoenauer, "Evolutionary algorithms for constrained engineering problems," *Computers & Industrial Engineering*, vol. 30, pp. 851-870, 1996.

[51] I. Osman, "u. Kelly, JP (Hrsg.): Meta-Heuristics. Theory and Applications," ed: Boston, MA: Springer US, 1996.

[52] J. Dréo, A. Pétrowski, P. Siarry, and E. Taillard, *Metaheuristics for hard optimization: methods and case studies*: Springer Science & Business Media, 2006.

[53] M. Gen and R. Cheng, "A survey of penalty techniques in genetic algorithms," in *Proceedings of IEEE international conference on evolutionary computation*, 1996, pp. 804-809.

[54] K. Krawiec, "Metaheuristic design pattern: Candidate solution repair," in *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 2014, pp. 1415-1418.

[55] D. Dasgupta and Z. Michalewicz, *Evolutionary algorithms in engineering applications*: Springer Science & Business Media, 2013.

[56] C. Grosan, A. Abraham, and M. Nicoara, "Search optimization using hybrid particle sub-swarms and evolutionary algorithms," *International Journal of Simulation Systems, Science & Technology*, vol. 6, pp. 60-79, 2005.

[57] M. Clerc, "Confinements and biases in particle swarm optimisation," 2006.

[58] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Applied mathematics and computation*, vol. 217, pp. 3166-3173, 2010.

[59] S.-M. Chen, A. Sarosh, and Y.-F. Dong, "Simulated annealing based artificial bee colony algorithm for global numerical optimization," *Applied mathematics and computation*, vol. 219, pp. 3575-3589, 2012.

[60] J. H. Holland and J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*: University of Michigan press, 1975.

[61] X.-S. Yang, *Engineering optimization: an introduction with metaheuristic applications*: John Wiley & Sons, 2010.
[62] R. L. Haupt and S. E. Haupt, *Practical genetic algorithms*: John Wiley & Sons, 2004.
[63] M. Thakur, "A new genetic algorithm for global optimization of multimodal continuous functions," *Journal of Computational Science,* vol. 5, pp. 298-311, 2014.
[64] Y. Del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, "Particle swarm optimization: basic concepts, variants and applications in power systems," *IEEE Transactions on evolutionary computation,* vol. 12, pp. 171-195, 2008.
[65] N. J. Radcliffe, "Equivalence class analysis of genetic algorithms," *Complex systems,* vol. 5, pp. 183-205, 1991.
[66] K. Deep and M. Thakur, "A new crossover operator for real coded genetic algorithms," *Applied mathematics and computation,* vol. 188, pp. 895-911, 2007.
[67] a. s. oussama, "fuzzy predictive control using meta-heuristic algorithms," doctorat, saad dahlab blida, 2015.
[68] C. Blum and D. Merkle, *Swarm intelligence: introduction and applications*: Springer Science & Business Media, 2008.
[69] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on evolutionary computation,* vol. 6, pp. 58-73, 2002.
[70] Z. Xinchao, "A perturbed particle swarm algorithm for numerical optimization," *Applied Soft Computing,* vol. 10, pp. 119-124, 2010.
[71] A. Silva, A. Neves, and E. Costa, "SAPPO: A simple, adaptable, predator prey optimiser," in *Portuguese Conference on Artificial Intelligence*, 2003, pp. 59-73.
[72] M. Nasir, S. Das, D. Maity, S. Sengupta, U. Halder, and P. N. Suganthan, "A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization," *Information Sciences,* vol. 209, pp. 16-36, 2012.
[73] S. He, Q. Wu, J. Wen, J. Saunders, and R. Paton, "A particle swarm optimizer with passive congregation," *Biosystems,* vol. 78, pp. 135-147, 2004.
[74] J. Sun, W. Fang, V. Palade, X. Wu, and W. Xu, "Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point," *Applied mathematics and computation*, vol. 218, pp. 3763-3775, 2011.
[75] B. O. Arani, P. Mirzabeygi, and M. S. Panahi, "An improved PSO algorithm with a territorial diversity-preserving scheme and enhanced exploration–exploitation balance," *Swarm and Evolutionary Computation,* vol. 11, pp. 1-15, 2013.
[76] W. Ji, J. Wang, and J. Zhang, "Improved PSO based on update strategy of double extreme value," *International Journal of Control and Automation,* vol. 7, pp. 231-240, 2014.
[77] H. Wang, H. Sun, C. Li, S. Rahnamayan, and J.-s. Pan, "Diversity enhanced particle swarm optimization with neighborhood search," *Information Sciences,* vol. 223, pp. 119-135, 2013.