الجمهوريـة الجزائـريـة الديمقراطيـة الشعبيـة
République Algérienne Démocratique et Populaire
وزارة التعليـم العالـي والبحـث العلمـي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

N° Réf :……

Centre Universitaire
Abd elhafid boussouf Mila

Institut des sciences et de la technologie **Département de Mathématiques et Informatique**

Mémoire préparé En vue de l'obtention du diplôme de Master

En: Mathématiques

Spécialité: Mathématiques appliquées

Algorithme d'optimisation chaotique de la mouche des fruits

Préparé par : Hasna Bouhamar

Amani Keziou

Soutenue devant le jury

Kaouache Smail	**MCA**	**C. U. Abdelhafid Boussouf, Mila**	**Président**
Bououden Rabah	**MCA**	**C. U. Abdelhafid Boussouf, Mila**	**Rapporteur**
Labed Boudjemaa	**MAA**	**C. U. Abdelhafid Boussouf, Mila**	**Examinateur**

Année universitaire :2021/2022

REMERCIEMENT

Au terme de ce travail, nous tenons à remercier Dieu le miséricordieux, qui nous a donné la force pour achever ce travail.

Nous tenons à remercier vivement notre promoteur Mr

"Rabah Bououden "

qui nous a proposé le thème de ce projet, pour sa disponibilité, son aide, ses précieux conseils. Nous sommes très heureux de pouvoir lui exprimer nos plus vifs remerciements et notre sincère reconnaissance, ses orientations et ses encouragements, tout au long de notre formation.

Nos vifs remerciements vont aux membres du jury qui nous ont honorés en acceptant d'évaluer et juger notre travail.

Nous remercions nos amis et les enseignants et les professeurs de Mathématiques et Informatiques , particulièrement ceux rattachés à la spécialité Mathematiques. à notre formation depuis notre premier cycle d'étude jusqu'à la fin de notre cycle.

Nous leurs exprimons notre profonde sympathie et nous leurs souhaitons une bonne continuation.

Dédicace

Avec les sentiments d'amour et de
gratitude Je dédie ce modeste travail :
A ma mère
et
A mon père .
source de force et courage .
A mes sœurs .
A mon frère.
A mes grands-parents .
A mes amies.

A tous ceux qui m'ont aidé et conseillé.

HASNA.

Dédicace

Je dédie ce modeste travail :

A ma mère .

A mon père.

A mon frère.

A ma sœur .

AMANI.

Résumé

L'objectif de ce mémoire est d'étudier l'algorithme d'optimisation chaotique de la mouche des fruits.

On va essayer de faire une étude détaillée des concepts dynamiques discrets Par exemple :Stabilité, bifurcation, Chaos...En plus nous présentons une brève introduction aux méthodes d'optimisation.

Enfin, nous avons testé l'algorithme d'optimisation de la mouche à fruits sur quatre problèmes standards (Sphère, Quartique, Somme des carrés,Rastrigin)en utilisation trois fonctions non linéaires différentes (Chebyshev , Iterative, Logistique)

Mots clés :

Algorithme d'optimisation chaotique, Bifurcation, Orbite périodique, Chaos.

Abstract

The objective of this memory is to study the chaotic optimization algorithm of the fruit fly. .

We first made a detailed study of the discrete dynamic concepts Such as :Stability, bifurcation,Chaos...In addition me present introduction to optimization methods.

Finally, we tested the fruit fly optimization algorithm on four standard problems (Sphere, Quartic, Sum squares,Rastrigin)by using three different nonlinear functions (Chebyshev ,Iterative ,Logistic)

Keywords :

Chaos optimization algorithm, Bifurcation, Periodic orbit, Chaos.

ملخص

الهدف من هذه المذكرة هو دراسة خوارزمية التحسين العشوائي لذبابة الفاكهة

حيث قمنا أولا بدراسة تفصيلية حول المفاهيم الديناميكية المنفصلة مثل: مفهوم الإستقرار، و تشعب، الفوضى ...هذا وبالإضافة إلى مقدمة موجزة لأساليب التحسين.

أخيرا إختبرنا خوارزمية التحسين الفوضوي لذبابة الفاكهة على أربعة مشاكل معيارية (كرة، رباعي، مجموع المربعات، راستريجن) بإستخدام ثلاثة وظائف غير خطية مختلفة (تشيبيشيف، ترابطي، لوجيستيك).

الكلمات المفتاحية: خوارزمية التحسين الفوضوي، كثافة الإحتمال، التشعب، فوضى .

Table des matières

Table des figures

INTRODUCTION GÉNÉRALE

En générale un système dynamique est un ensemble d'objet en interaction au cours du temps. Pour l'étude des système dynamique il faut connaitre qu'ils existent des systèmes à dynamique régulier et d'autre à dynamique chaotique, des systèmes très simples et d'autre très complexes

Un peut d'historique sur le chaos [1]

- En 1890 Le Roi Oscar II de Suède octroie un prix au premier chercheur qui pourrait déterminer et résoudre le problème des n-corps des orbites des corps célestes et ainsi prouver la stabilité du système solaire. Jusqu'à ce jour, le problème n'a pas été résolu.
- En 1890 Henri Poincaré gagne le premier prix du Roi Oscar II. Etant le plus proche à résoudre le problème des n-corps, il a découvert que l'orbite de trois corps célestes agissantes l'une sur l'autre peut engendrer un comportement instable et imprévisible. Ainsi, le chaos est naît (mais pas encore mentionné !).
- En 1963 Edward Lorenz découvre le premier système chaotique dans la météo ou encore appelé attracteur étrange [2].
- En 1975 Tien-Yien Li et James A. Yorke ont introduit pour la première fois le terme "chaos" dans un article intitulé "Period three implies chaos" [3].
- En 1978 Mitchell Feigenbaum introduit un nombre universel associé au chaos.

Revenons au mot optimisation qui est le deuxième axe de cette mémoire. Pour

les problèmes d'optimisation de certaines fonctions usuelles qui sont différentiables, certains algorithmes d'optimisation classiques tels que la méthode de Newton, la méthode du gradient et la méthode Hessians [4, 5], peuvent obtenir leurs points globaux optimaux avec l'avantage d'une vitesse de convergence et de la haute précision. Cependant, ces algorithmes d'optimisation traditionnels converges facilement à optimum local lors de la résolution des problèmes d'optimisation de certaines fonctions multi-dimensionnelles.

Récemment, les chercheurs se sont concentrés sur le développement d'algorithmes hybrides en combinant des algorithmes heuristiques avec les techniques de recherche chaotiques pour résoudre des systèmes d'équations non linéaires et des problèmes d'optimisation tels que l'optimisation de Monte Carlo chaotique, laBFGS chaotique, l'optimisation chaotique d'essaims de particules, les algorithmes génétiques chaotiques, recuit simulé chaotique, etc [6–8].

Ce mémoire est composée de trois chapitres : Dans le **chapitre 1**, nous présentons des notions de base des systèmes dynamiques discrets (notions, définitions, théorèmes, et critères de stabilité).
Dans le **chapitre 2**, nous donnons une l'introduction à l'optimisation. Dans le **dernier chapitre** nous étudions Algorithme d'optimisation chaotique de la mouche des fruits .
on termine ce travail par une conclusion générale .

Chapitre 1

Préliminaire sur les systèmes dynamiques non linéaires discrets

Le but de ce premier chapitre est de présenter quelques rappelles sur les systèmes dynamiques et leurs propriétés générales.

1.1 Système dynamique discret

Un système dynamique discrets est représenté par une équation aux différences finies comme suit :

$$x(k+1) = F(x(k), k), \tag{1.1}$$

où

$$x(k) \in \mathbb{R}^n, K \in \mathbb{N} \; et \; F : \mathbb{R}^n \times \mathbb{N} \to \mathbb{R}^n.$$

On peut écrire aussi :

$$F^0 = x; F^1 = F(x); F^2 = F(F(x)); ...F^k(x) = F(F^{k-1}(x)),$$

et

$$x_0, x_1 = F(x_0), x_2 = F^2(x_0), ..., x_k = F^k(x_0),$$

1.1.1 Orbites

L'orbite de x par le système dynamique F est défine par :

$$O = \{F^k(x), K \in \mathbb{N}\}, \tag{1.2}$$

1.1.2 Points fixes

On appelle "point fixe" d'un système dynamique discret F tout point x tel que :

$$F(x) = x, \tag{1.3}$$

Parfois, ces points sont appelés aussi points stationnaires ou points d'équilibre.Si la matrice jacobienne DF(x) n'a pas de valeurs propres dont le module soit égal à +1, x est un point fixe hyperbolique. Si tous les modules des valeurs propres de DF(x) sont égaux à +1, x est un point fixe elliptique.

1.1.3 Points périodiques et p-cycles

S'il existe $k \geq 1$, tel que $F^k(x) = x$, on dit que x est un point périodique. La période d'un point périodique x est le plus petit entier $k \geq 1$ tel que :

$$F^k(x) = x, \tag{1.4}$$

Un ensemble $\{x_0, x_1, ...x_{p-1}\}$ forme un cycle d'ordre p (ou une orbite périodique d'ordre p, ou encore un p cycle), si :

$$\begin{cases} F(x(i)) & = & x(i+1), i = 1, ..., p-1. \\ F(x(p-1)) & = & x(0). \end{cases} \tag{1.5}$$

Autrement dit, chaque point d'un cycle d'ordre p est un point fixe pour F^p où $F^p(x(i)) = x(i)$; pour i = 0 ; 1 ; ... ; p-1. et n'est pas un point fixe pour F^k

si $k < p$.

1.2 Stabilité

L'étude du comportement d'un système dynamique discret, correspond à l'étude de stabilité des points fixes. Nous n'abordons ici que le problème du point fixe et pour les points périodiques de période p, il suffit de considérer la p-ième itérée de l'application. Soit le système dynamique non linéaire :

$$x(k+1) = F(x(k)), \tag{1.6}$$

Dont la réponse est telle que :

$$x(k) = F(k, k_0, x(k_0)), \tag{1.7}$$

Et les condition initiales définies par

$$x(k_0) = x(0), \tag{1.8}$$

Soit x_f un point fixe du système, on a :

$$x_f = F(x_f), \tag{1.9}$$

1.2.1 Définitions

Définition 1.2.1. *Le sytème est dit stable au sens de Lyapunov par rapport au point fixe x_f si pour des conditions initiales $x(k_0)$ suffisament proches du point fixe soit :*

$$\forall \epsilon > 0, \exists \delta : \| x(k_0) - x_f \| < \delta \Rightarrow \| x(k, k_0, x(k_0)) - x_f \| < \epsilon, \forall k \geq k_0. \tag{1.10}$$

Définition 1.2.2. *Le point fixe x_f est attractif lorsqu'il y a convergence de l'état x vers l'état x_f au bout d'un temps infini, les conditions initiales $x(k_0)$ étant bornées, soit :*

$$\forall k_0 \in \mathbb{N}; \exists \delta_0(k_0), telque : \|x(k_0 - x_f)\| < \delta_0(x_0) \Rightarrow limx(k, k_0, x(k_0)) = x_f,$$

Lorsque $\delta_0(k_0) = +\infty$ *; on dit que le point fixe* x_f *est globalement attractif.*

Définition 1.2.3. *Le point fixe* x_f *est dit asymptotiquement (respectivement globalement asymptotiquement) stable lorsqu'il est à la fois stable au sen de Lyapunov et attractif (respectivement globalement asymptotiquement)*

1.2.2 Linéarisation

OU considère le cas où le systéme non linéaire décrit par (1.6) admet, au voisinage de $x_f = 0$ un développement limité de la forme :

$$x(k+1) = Ax(k) + r(\|x\|), \tag{1.11}$$

dont lequel la matrice A est constante est :

$$\lim_{x \to 0} \frac{\| r(\| x \|) \|}{\| x \|} = 0. \tag{1.12}$$

Le système linéaire est décrit par la relation :

$$x(k+1) = Ax(k); \tag{1.13}$$

peut être considéré comme la linéarisation de (1.6) autour $x_f = 0$.il permet de statuer, localement, sur la stabilité du système non linéaire au point $x_f = 0$

1.2.3 Fonction de Lyapunov

La deuxième méthode de Lyapunov permet d'étudier l'analyse de la stabilité directement à partir des équations qui décrivent le système et ne nécessitent pas la détermination explicite de leurs solutions. Nous introduisons une fonction continue

$v(x(k)) : R^n \to R^+$, dite de Lyapunov, vérifiant : v (x (k)) définе positive, c'est-à-dire $v(x(k)) > 0; \forall x(k) \neq 0; etv(0) = 0$.

1.3 Bifurcation

1.3.1 Définitions

Soit le système dynamique non linéaire suivant

$$x(k+1) = F(x(k), \alpha), \tag{1.14}$$

D'où

$$x(k) \in \mathbb{R}^n, \alpha \in \mathbb{R}^m, k \in \mathbb{N} etF : \mathbb{R}^n * \mathbb{R}^m \to \mathbb{R}^n.$$

Définition 1.3.1. *Une bifurcation est un changement qualitatif de la solution x_f du système (1.15) lorsqu'on modifie le paramètre de contrôle, c'est à dire la disparition ou le changement de stabilité et l'apparition de nouvelles solutions.*

Définition 1.3.2. *Un diagramme de bifurcation est une portion de l'espace des paramètres sur laquelle sont représentes tous les points de bifurcation.*

1.3.2 Types de bifurcations

Il existe plusieurs types de bifurcation selon les propriétés des secondes dérivées de la famille des fonction $F(x(k), \alpha)$. Chacune de ces bifurcations est caractérisée par une forme normalequi est l'équation générale typique de ce type de bifurcation [9,10 ,11]. . Parmi les différents types de bifurcations, pour les systémes dynamiques discrets, on trouve [12] :

1-**Bifurcation de type noed-col(ou tangente,ou pli)** : Cette bifurcation se produit lorsque l'une des deux valeurs propres de $DF(x(k), \alpha)$ est égale à $+1$:

sur le diagramme des bifurcations on observe, dans ce cas, une courbe de points fixes continue tangente à la ligne droite verticale.Deux points d'équilibres existent (un stable et un instable) avant la bifurcation. Après la bifurcation, plus aucun équilibre n'existe.

2-**Bifurcation transcritique** : Sur le diagramme de bifurcations cela se traduit par deux branches différentes de points fixes qui se croisent en un point et par le changement de stabilité des deux branches au passage par le point d'intersection.

3-**Bifurcation de doublement de période (ou flip)** : Cette bifurcation a lieu lorsque l'une des deux valeurs propres de $DF(x(k), \alpha)$ est égales à -1. Un point fixe stable d'ordre 1, devient instable en même temps que l'apparition d'un cycle d'ordre 2 stable.

4-**Bifurcationde Neimark-Sacker** : Cette bifurcation se produit lorsque $DF(x(k); \alpha)$ possède deux valeurs propres complexe egale à $\exp(\pm i\theta)$

1.4 Généralités sur le chaos

1.4.1 Définition de chaos

On trouve dans la littérature plusieurs définitions mathématiques du chaos, mais jusqu'à présent, il n'existe aucune définition mathématique universelle du chaos. Avant de donner la définition du chaos, due à R.L Devaney, quelques définitions de base sont [13, 14]. . Soit $(I \subset \mathbb{R}, d)$ désignant un espace métrique compact (d est une distance), et soit F la fonction

$$F : I \to I, x(k+1) = F(x(k)), x(0) \in I.$$

Définition 1.4.1. *Supposons que X est un ensemble et Y un sous-ensemble de X. Y est dense dans X si, pour n'importe quel élémentx* $\in X$*, il existe un élément y dans le sous-ensemble Y arbitrairement proche de x, c'est-à-dire si la fermeture de Y est égale à* $X(Y = X)$*. Ce qui revient à dire que Y est dense dans X si pour tout* $x \in X$ *on peut trouver une séquence de points* $y_n \in Y$ *qui convergent vers x .*

Définition 1.4.2. *F est dite avoir la propriété de sensibilité aux conditions initiales s'il existe* $\delta > 0$ *tel que, pour* $x(0) \in I$ *et pour tout* $\varepsilon > 0$ *il exist un point* $y(0) \in I$ *point et un entier* $j \geq 0$ *satisfaisant :*

$$d(x(0), y(0)) > \varepsilon \Rightarrow d(F^j(x(0)), F^j(y(0))) > \delta$$

,
Où d représente la distance et F^j *la j ème itération de F.*

Définition 1.4.3. *F est dite topologiquement transitive si U et V étant deux ensembles non vides ouverts dans I , il exist* $x(0) \in U$ *et un indice* $j \in Z^+$ *,tel que pour* $F^j(x(0)) \in V$ *ou de façon équivalente, il exist un indice* $j \in Z^+$ *tel que pour* $F^j(U) \cap V \neq 0$ *On est maintenant en position d'énoncer la défintion du chaos, au sens de* ***Devany*** *[15].*

Définition 1.4.4. *La fonction F (2.1) est dite constituée d'une dynamique chaotique si :*
(i) F possède une sensibilité aux conditions initiales.
(ii) F est topologiquement transitive.
(iii) L'esemble des points périodiques de F est denses dans I.
Bien qu'il n'existe pas de définition universellement acceptée de la notion du chaos, cette définition reste la plus intéressante car les concepts sur lesquels elle repose sont facilement observables.

1.4.2 Caractéristique du chaos

Sensibilité aux condition initiales

La sensibilité aux conditions initiales est un phénomène découvert pour la première fois, dès la fin du xixe siècle par **Poincaré**, puis a été redécouvert en 1963

par Lorenz lors de ses travaux en météorologie. Cette découverte a entrainé un grand nombre de travaux importants, principalement dans le domaine des mathématiques. Cette sensibilité explique le fait que, pour un système chaotique, une modification infime des conditions initiales peut entrainer des résultats imprévisibles sur le long terme. Le degré de sensibilité aux conditions initiales quantifie caractère chaotique du système [16,17]..

Exposants de Lyapunov

Les exposants de Lyapunov servent à mesurer la divergence possible entre deux orbites issues de conditions initiales voisines et permettent de quantifier la sensibilité aux conditions initiales d'un système chaotique. Le nombre des exposants de Lyapunov est égal à la dimension de l'espace des phases [18,19, 20] . Soit le système dynamique nonlinéaire discret suivant :

$$x(k+1) = F(x(k)), \tag{1.15}$$

avec $x(k) \in \mathbb{R}^n$ Nous supposons que la trajectoire émanant d'un état initial $x(0)$ atteint un attracteur. $x(k)$ est ainsi bornée à l'interieur de l'attracteur. Nous choisissons deux conditions initiales très proches, note $x(0)$ et $x(0)$ et ne regardons comment se comportent les trajectoires qui en sont issues. En supposant que les deux trajectiores $x(k)$ et$x(k)$ s'ecart expontialement, apres k il vient :

$$|\dot{x}(k) - x(k)| = |\dot{x}(0) - x(0)|e^{\lambda k} \tag{1.16}$$

λ indique le taux de divergence par itération des deux trajectoires dont l'expression est la suivante :

$$\lambda = \frac{1}{k} \ln \left| \frac{\dot{x}(k) - x(k)}{\dot{x}(0) - x(0)} \right| \tag{1.17}$$

Pour $x(0)$ et $x(0)$ proche, si le module de la différence $\varepsilon = |\dot{x}(0) - x(0)|$ a tendance à converger ver zéro, on obtient :

$$\lambda_L = \lim_{k \to \infty} \frac{1}{k} \lim_{\varepsilon \to 0} \ln \left| \frac{\dot{x}(k) - x(k)}{\dot{x}(0) - x(0)} \right| \tag{1.18}$$

Cela donne :

$$\lambda_L = \lim_{k\to\infty}\lim_{\varepsilon\to 0}\frac{1}{k}\ln\left|\frac{\dot{x}(k)-x(k)}{\dot{x}(k-1)-x(k-1)} * \frac{\dot{x}(k-1)-x(k-1)}{\dot{x}(k-2)-x(k-2)} * ... * \frac{\dot{x}(1)-x(1)}{\dot{x}(0)-x(0)}\right|$$

$$= \lim_{k\to\infty}\lim_{\varepsilon\to 0}\frac{1}{k}\sum_{i=0}^{k-1}\ln\left|\frac{\dot{x}(i+1)-x(i+1)}{\dot{x}(i)-x(i)}\right|$$

$$= \lim_{k\to\infty}\lim_{\varepsilon\to 0}\frac{1}{k}\sum_{i=0}^{k-1}\ln\left|\frac{F(\dot{x}(i))-F(x(i))}{\dot{x}(i)-x(i)}\right|$$

Finalement ona :

$$\lambda_L = \lim_{k\to\infty}\lim_{\varepsilon\to 0}\frac{1}{k}\sum_{i=0}^{k-1}\ln\left|\frac{dF(x(i))}{dx(i)}\right|$$

λ_L, appelé exposant de Lyapunov, mesure le taux moyen de divergence de deux trajectoires distinctes, à partir de deux conditions initiales très proches. Dans le cas d'un systeme de dimension $n > 1$il exist n exposant de Lyapunov
$\lambda_L^{(j)}, (j = 1, 2, ..., n)$, chacun d'entre eux mesure le taux de divergence suivant un des axes de l'espace de phase. Pour le calcul de l'exposant de Lyapunov, nous partons d'un point initial $x(0) \in \Re^n$ pour caractériser le comportement infinitisémal autour du point x(k) par la première matrice dérivée $DF(x(i))$

$$DF(x(i)) = \begin{pmatrix} \frac{\partial f_1(x(i))}{\partial x_1(i)} & \cdots & \frac{\partial f_1(x(i))}{\partial x_n(i)} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_1(x(i))}{\partial x_1(i)} & \cdots & \frac{\partial f_n(x(i))}{\partial x_n(i)} \end{pmatrix}$$

Notons : $J_k = DF(x(k-1))...DF(x(0)), avec : J_0 = DF(x(0))$. L'exposant de Lyapunov est calculé par l'expression suivante :

$$\lambda_L^{(j)} = \lim_{k\to\infty}\frac{1}{k}\ln\left|\lambda_i(J_k...J_1)\right|, i = 1, 2, ..., n$$

En analysant les exposants de Lyapunov d'un système, nous pouvons conclure sur le type du comportement de ce système comme suit :

- **Si** $\lambda_n \leq ... \leq \lambda_1 < 0$,il existe des point fixes asymptotiquement stables.

- **Si** $\lambda_1 = 0, \lambda_n \leq ... \leq \lambda_2 < 0$,l'attracteur est un cycle limite asymptotiquement stable.
- **Si** $\lambda_1 = ... = \lambda_k = 0, \lambda_n \leq ... \leq \lambda_{k+1} < 0$,l'attracteur est un tore de dimension k, c'est-à-dire quasi-periodique.
- **Si** $\lambda_1 > 0, \sum_i \lambda_i < 0$,l'attracteur est chaotique.
- **Si** $\lambda_1 > ... > \lambda_k > 0, \sum_i \lambda_i < 0$,l'attracteur est hyperchaotique.

Dimention fractale

Il existe plusieurs types de dimensions fractale (dimension de capacité, dimension d'information, dimension de corrélation,...) pour les attracteurs chaotiques, parmi celle-ci on peut citer :

Dimention de Hausdorff

La dimension de Hausdorff de $M \in \mathbb{R}^n$ est définit par[21] :

$$D_H = sup\{d, \mu_d(M) = +\infty\} = inf\{d, \mu_d = 0\}$$

D'où $\mu_d(M)$ est la mesure d-dimensionnelle de Hausdorff de l'ensemble M : Ce type de dimension dépend uniquement des propriétés métriques de l'espace dans lequel se trouve l'ensemble (attracteur ou non).

Dimention de Lyapunov

la dimention de Lyapunov est donné par [22] :

$$D_L = \frac{\sum_{i=1}^{j} \lambda_i}{|\lambda_{j+1}|} + j$$

D'où $\lambda_n \leq ... \leq \lambda_1$sont les exposants de Lyapunov d'un attracteur d'un système dynamique et j le grand entier naturel tel que : $\sum_{i=1}^{j} \lambda_i \geq 0$. Ce type de dimensions tient compte de la dynamique du système .

Attracteur étrange

L'attracteur étrange est une caractérstique géometrique du chaos. Il n'existe pas une définition rigoureuse d'un attracteur étrange ou chaotique et toutes les définitions qui on trouve dans la ittérature sont restrictives [23,24,25 ,26 ,27].

Définition Un sous-ensemble borné A de l'espace des phases est un attracteur étrange pour une transformation T de l'espace s'il existe un voisinage U de A, c'est à dire que pour tout point de A il existe une boule contenant ce point et contenue dans $\mathbb{R}$ vérifiant les propriétés suivantes :

1) U est une zone de capture, ce qui signifie que toute orbite par T dont le point initial est dans U est entièrement contenue dans U . De plus, toute orbite de ce type devient et reste aussi proche de A que l'on veut.

2) Les orbites dont le point initial est dans $\mathbb{R}$ sont extrêmement sensibles aux conditions initiales.

3) A est un objet fractal.

4) Pour tout point de A, il existe des orbites démarrées dans $\mathbb{R}$ qui passent aussi prés que l'on veut de ce point.

1.5 Exemples de systèmes chaotiques discrets (SCD)

1.5.1 Systèmes chaotiques discrets dans le plan

Attracteur d'Hénon

L'attracteur d'Hénon [20], est un système dynamique discret de dimension 2 dont la représentation d'état est la suivante :

$$\begin{cases} x_1(k+1) &= a - x_1{}^2(k) + bx_2(k). \\ x_2(k+1) &= x_1(k). \end{cases} \tag{1.19}$$

Pour les valeurs a = 1,4 et b = 0,3, cet attracteur présente un comportement chaotique, comme l'illustre la figure 1.1.

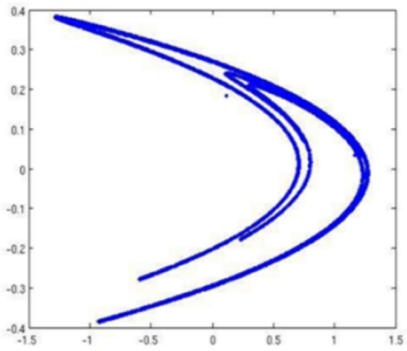

FIGURE 1.1 – *L'attracteur chaotique d'Hénon discret pour les valeurs* $a = 1.25$ *et* $b = 0.75$.

Système de Lorenz discret

Le système de Lorenz discret [28], est donné par :

$$\begin{cases} x_1(k+1) &= (1+\alpha\beta)x_1(k) - \beta x_2(k)x_1(k). \\ x_2(k+1) &= (1-\beta)x_2(k) + \beta x_1^2. \end{cases} \tag{1.20}$$

1.5.2 Systèmes chaotiques discrets dans l'espace

Système de Hitzl-Zele

Hitzl et Zele [29], obtiennent le système généralisé d'Hénon

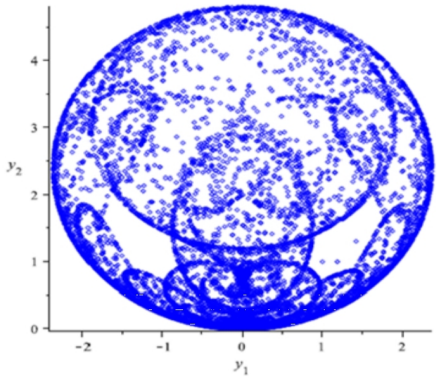

FIGURE 1.2 – *L'attracteur chaotique de Lorenz pour les valeurs* $a = 1.4$ *et* $b = 0.3$.

$$\begin{cases} x_1(k+1) &= -\beta x_2(k), \\ x_2(k+1) &= x_3(k) + 1 - \alpha x_2^2(k), \\ x_3(k+1) &= \beta x_2(k) + x_1(k), \end{cases} \tag{1.21}$$

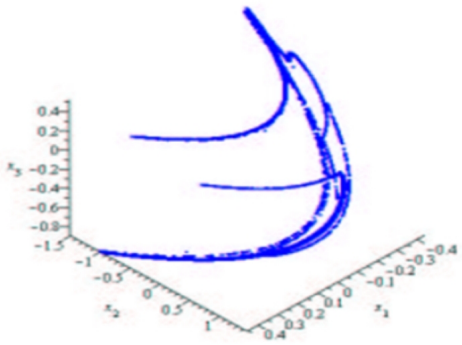

FIGURE 1.3 – *L'attracteur chaotique d'Hitzl-Zele lorsque* $(\alpha, \beta) = (1.07, 0.3)$.

Système de Rössler discret

Le système discret de Rössler [30], est représenté par :

$$\begin{cases} x_1(k+1) &= \alpha x_1(k)(1-x_1(k)) - \beta(x_3(k)+\gamma)(1-2x_2(k)), \\ x_2(k+1) &= \delta x_2(k)(1-x_2(k)) + \varsigma x_3(k), \\ x_3(k+1) &= \eta(1-\theta x_1(k))[(x_3(k)+\gamma)(1-2x_2(k))-1], \end{cases} \tag{1.22}$$

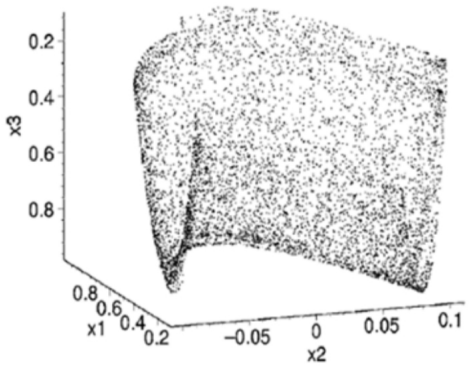

FIGURE 1.4 – *L'attracteur hyperchaotique du système discret de Rossler.*

Tels que $\alpha = 3.8, \beta = 0.05, \gamma = 0.35, \delta = 3.78, \varsigma = 0.2, \eta = 0.1, \theta = 1.9$. L'attracteur hyperchaotique du système (1.23) est représenté dans la Figure 1.5

Système de Wang

Le système de Wang [31], est décrit comme suit

$$\begin{cases} x_1(k+1) &= a_3 x_2(k)(a_4+1)x_1(k), \\ x_2(k+1) &= a_1 x_1(k) + y_2(K) + a_2 x_3(k), \\ x_3(k+1) &= (a_7+1)x_3(k) + a_6 x_2(k) x_3(k) + a_5, \end{cases} \tag{1.23}$$

Le système de Wang est hyperchaotique lorsque les valeurs des paramètres de bifurcation sont considérés comme $(a_1, a_2, a_3, a_4, a_5, a_6, a_7) = (1.9, 0.2, 0.5, 2.3, 2, 0.6, 1.9)$. L'attracteur hyperchaotique du système de Wang est présenté dans la Figure 1.5.

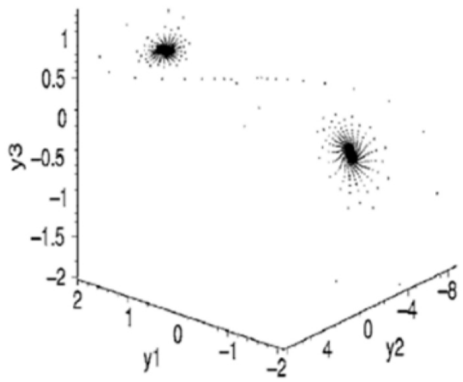

FIGURE 1.5 – *L'attracteur hyperchaotique de Wang lorsque* $(a_1, a_2, a_3, a_4, a_5, a_6, a_7) = (-1.9, 0.2, 0.5, -2.3, 2, -0.6, -1.9)$.

Chapitre 2

Introduction à l'optimisation

2.1 Types d'optima

Il y a plusieurs types de points optimum parmi eux

1. Optima locaux et globaux Si le point x^*dans le domaine de définition D_f de la fonction f vérifié :

$$f(x) \geq f(x^*), \forall x \in D_f$$

On dit que x^* est un optimum global de la fonction f . D'autre part s'il existe un $\epsilon > 0$ tel que :

$$f(x) \geq f(x^*), \forall x \in D_f \cap V_\epsilon(x^*)$$

Où $V_\epsilon(x^*)$ dénote un voisinage de diamètre ϵ centré en x^* ;alors x^* est dite optimum locale de la fonction f (Figure 3.1).

2. Optima stricts

Définition 3.1

Un minimum local x^*est dit strict s'il existe une valeur $\epsilon > 0$ telle que quel que soit $x \in V_\epsilon(x^*), x \neq x^*, f(x) > f(x^*)$, (l'inégalité est stricte).

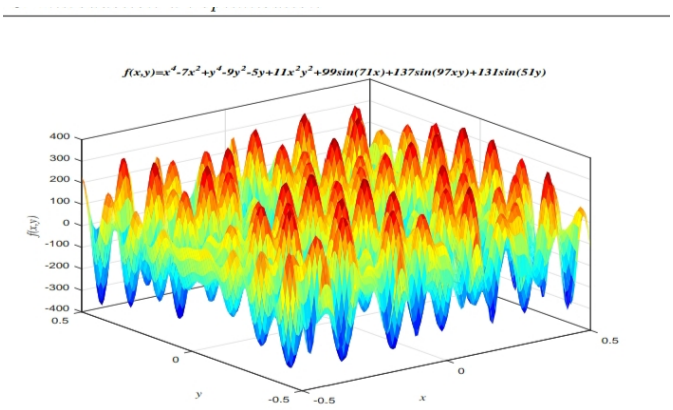

FIGURE 2.1 – *Fonction comportant plusieurs minima locaux près de l'origine (mais il est très difficile d4identifier son optimum globale)..*

Nous discutons dans les deux sections suivantes deux classes des méthodes d'optimisation, la première classe inclue les méthodes d'optimisation classiques (méthode de la section dorée, méthode de Fibonacci,...) ; la deuxième classe inclue les méthodes méta-heuristiques (optimisation par essaims particulaires, optimisation des colonies de fourmis,...).

2.2 Méthodes d'optimisation classiques

Les méthodes classiques d'optimisation sont utiles pour trouver la solution optimale des fonctions continues et différentiables. Ces méthodes sont analytiques et utilisent les techniques du calcul différentiel pour localiser les points optimaux.

2.2.1 Optimisation unidimensionnelle

1 Méthode de la section dorée

Cette méthode est valable uniquement pour des fonctions à valeurs réelles,

dont on connait un intervalle $[a_0, b_0]$ sur lequel elle admet un unique minimum x^* figure(2.2). Considérons une fonction unidimensionnelle f et l'intervalle $[a_0, b_0]$ Cette méthode consiste à évaluer f à deux points intermédiaires, comme illustré à la figure (3.3). Nous choisissons les points intermédiaires de telle sorte que la réduction

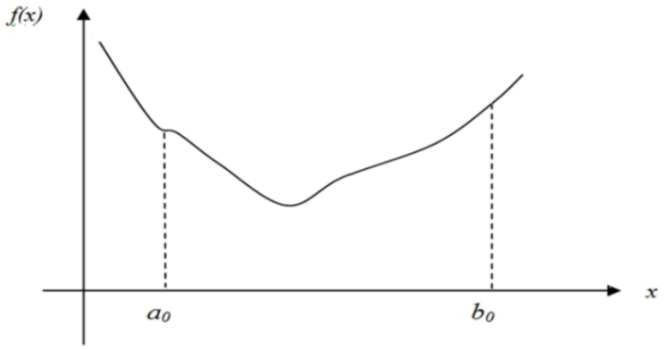

FIGURE 2.2 – *Graphe d'une fonction réelle présentant un unique minimum sur un intervalle* $[a_0, b_0]$.

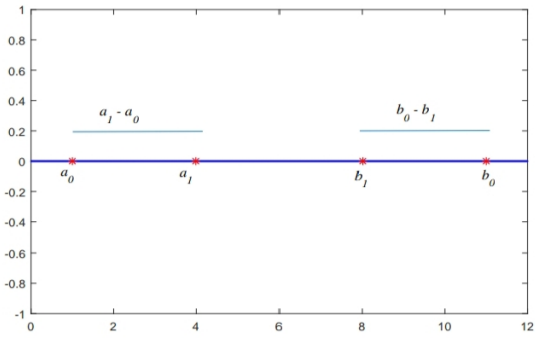

FIGURE 2.3 – *Méthode de la section dorée.*

de l'intervalle soit symétrique ,en sens que :

$$a_1 - a_0 = b_0 - b_1 = \rho(b_0 - a_0)$$

Avec $\rho < \frac{1}{2}$. Nous évaluons ensuite f aux points intermédiaires. Si $f(a_1) < f(b_2)$, x^* doit être compris dans l'intervalle $[a_0, b_1]$ Si, au contraire,$f(a_1) > f(b_1)$, x^* est situé dans l'intervalle $[a_1, b_0]$. Nous pouvons répéter le processus et trouver deux nouveaux points, par exemple $a_2$et $b_2$en utilisant la même valeur de $\rho < \frac{1}{2}$ comme avant. Bien sûr nous aimerions minimiser le nombre d'évaluations de la fonction objective tout en réduisant la largeur de l'intervalle d'incertitude. Supposons, par exemple, que $f(a_1) < f(b_1)$ Donc, nous savons que $x^* \in [a_0, b_1]$. Puisque $a_1$est déjà dans l'intervalle d'incertitude et que $f(a_1)$ est déjà connu, nous pouvons faire coïncider un a_1 avecb_2 . Ainsi, une seule nouvelle évaluation de f en a_2 serait nécessaire. Nous devons trouver la valeur de ρ pour laquelle nous évaluons f une seule fois. Sans perte de généralité, nous supposons que l'intervalle $[a_0, b_0]$ est de longueur unitaire. Ensuite, pour avoir une seule nouvelle évaluation de f il suffit de choisir ρ pour que :

$$\rho(b_1 - a_0) = b_1 - b_2$$

Parce que $b_1 - a_0 = 1 - \rho$ et $b_1 - b_2 = 1 - 2\rho$, on a

$$\rho(1 - \rho) = 1 - 2\rho$$

D'où

$$\rho^2 - 3\rho + 1 = 0$$

Les solutions sont $\rho_1 = \frac{3+\sqrt{5}}{2}$,$\rho_2 = \frac{3-\sqrt{5}}{2}$ et comme $\rho < \frac{1}{2}$, nous prenons
$\rho = \frac{3-\sqrt{5}}{2} = 0.382$. Remarquons que $1 - \rho = \frac{\sqrt{5}-1}{2}$, donc

$$\frac{\rho}{1-\rho} = \frac{1-\rho}{1}$$

2 Méthode de Fibonacci

Rappelons que la méthode de la section dorée utilise la même valeur de ρ partout. Supposons maintenant que nous sommes autorisés à faire varier la valeur ρ d'une étape à l'autre, de sorte qu'à la k^{me} étape nous utilisions une valeur ρ_k, et à l'étape suivante, nous utilisions une valeur ρ_{k+1} , etc. Comme dans la méthode de la section dorée, notre objectif est de sélectionner les valeurs successives de ρ_k, $0 \leq \rho_k \leq \frac{1}{2}$, de sorte qu'une seule nouvelle évaluation de la fonction f est requise à chaque étape. Pour dériver la stratégie de sélection des points d'évaluation, considérons la figure (3.4). On voit qu'il suffit de choisir le ρ_k tel que

$\rho_{k+1}(1-\rho_k) = 1-2\rho_k$

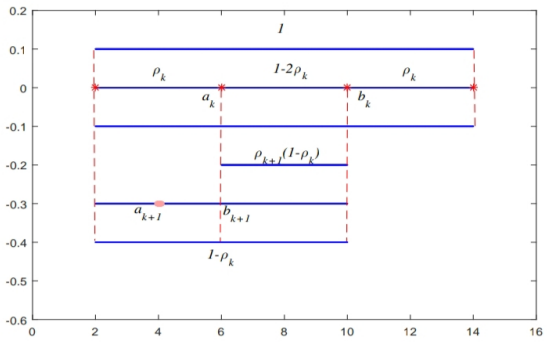

FIGURE 2.4 – *Méthode de Fibonacci.*

D'où

$$\rho_{k+1}(1-\rho_k) = 1-2\rho_k \tag{2.1}$$

Il existe de nombreuses suites ρ_1 ;ρ_2 ; ... qui satisfont l'équation (2.1) et la condition $0 \leq \rho_k \leq \frac{1}{2}$.Par exemple, la séquence ρ_1=ρ_2=ρ_3=...= $\frac{3-\sqrt{5}}{2}$ satisfait les conditions ci-dessus et donne lieu à la méthode de la section dorée. Supposons que nous obtenions une suite ρ_1 ;ρ_2 ; ... qui satisfait les conditions ci-dessus et que nous

utilisions cette suite dans notre algorithme de recherche. Donc, après N itérations de l'algorithme, l'intervalle d'incertitude est réduite d'un facteur de

$(1-\rho_1)(1-\rho_2)...(1-\rho_N)$. En fonction de la suite ρ_1 ;ρ_2 ; ... , nous obtenons un facteur de réduction différent. La question naturelle est la suivante : Quelle suite ρ_1 ;ρ_2 ; ... minimise le facteur de réduction ci-dessus ? Ce problème est un problème d'optimisation avec contrainte qui peut être formulé comme suit : minimiser $(1-\rho_1)(1-\rho_2)...(1-\rho_N)$. Tel que $\rho_{K+1} = 1 - \frac{\rho_k}{1-\rho_k}$; k = 1, 2, ...,N-1. Et $0 \leq \rho_k \leq \frac{1}{2}$, k = 1, 2, ...,N-1.

Avant de donner la solution au problème d'optimisation ci-dessus, nous devons d'abord introduire la suite de Fibonacci,F_1, F_2,... Cette suite est définie comme-suit : posons $F_{-1} = 0$ et $F_0 = 1$ par convention. Puis pour $k \geq 0$ on a

$F_{k+1} = F_k + F_{k-1}$

Certaines valeurs des éléments de la suite de Fibonacci sont les suivantes : 1, 2, 3, 5, 8, 13, 21, 34. Il s'avère que la solution au problème d'optimisation ci-dessus est la suivante :

$$\begin{aligned}
\rho_1 &= 1 - \tfrac{F_N}{F_{N+1}} \ , \\
\rho_2 &= 1 - \tfrac{F_{N-1}}{F_N} \ , \\
&\vdots \\
\rho_K &= 1 - \tfrac{F_{N-K+1}}{F_{N-K+2}}. \ , \\
&\vdots \\
\rho_N &= 1 - \tfrac{F_1}{F_2}.
\end{aligned}$$

où F_1, F_2 sont les éléments de la suite de Fibonacci. L'algorithme résultant est appelé méthode de recherche de Fibonacci. Pour la preuve de l'optimalité de la méthode de recherche de Fibonacci voir [32]. Pour une étude approfondie d'autres types de méthodes d'optimisation unidimensionnelle (méthode de Newton, méthode de Secant,...) voir [33, 34]. Nous présentons dans la sous section suivante

quelques méthodes d'optimisation classiques multidimensionnels

2.2.2 Optimisation multidimensionnelle

1 Méthode de Gradient

Rappelons qu'une surface de niveau d'une fonction est l'ensemble $F : \mathbb{R}^n \to \mathbb{R}$ des points x satisfaisant $f(x) = c$ pour une constante c.

Le gradient de la fonction f en x_0,noté $\triangledown f(x_0)$,s'il n'est pas un vecteur nul, est orthogonal au vecteur tangent à une courbe lisse arbitraire passant par x_0 au surface de niveau f (x) = c. Ainsi, la direction du taux de croissance maximal d'une fonction différentiable à valeurs réelles en un point est orthogonale au surface de niveaux de la fonction passant par ce point.

Ainsi, la direction dans laquelle $\triangledown f(x_0)$ pointe est la direction du taux de croissance maximal de f en x. La direction dans laquelle $\triangledown f(x_0)$ pointe est la direction du taux maximum de décroissance de f en x. Par conséquent, la direction du gradient négatif est une bonne direction de recherche si nous voulons trouver un minimum de la fonction f . Basé sur ce qui précède, soit un point de départ x_0 et considérons le point $x_0 - \alpha \nabla f x_0$. Ensuite, par le théorème de Taylor, nous obtenons

$$f(x_0 - \alpha \nabla f x_0) = f(x_0) - \alpha \parallel \nabla f x_0 \parallel^2 + o(\alpha) \tag{2.2}$$

Ainsi, si $f(x_0) \neq 0$, alors pour un $\alpha > 0$ assez petit, nous avons

$$f(x_0 - \alpha \nabla f x_0) < f(x_0) \tag{2.3}$$

Cela signifie que le point $x_0 - \alpha \nabla f x_0$ est une amélioration par rapport au point x_0 si nous cherchons un minimum de la fonction f . Pour formuler un algorithme qui implémente l'idée ci-dessus, supposons qu'un point x_k soit donné. Pour trouver le prochain point x_{k+1}, nous commençons à x_k et nous déplaçons d'un montant

$-\alpha_k \nabla f(x_k)$, où α_k est un scalaire positif appelé le pas. La procédure ci-dessus conduit à l'algorithme (algorithme de gradient) itératif suivant :

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) \tag{2.4}$$

Nous avons le choix de prendre de très petites pas et de réévaluer le gradient à chaque étape, ou nous pouvons faire de grands pas à chaque fois.

*** Méthode du gradient à pas constant** On utilise le plus souvent la méthode du gradient à pas constant $\alpha_k = \alpha$. Comme on peut varier le pas α_k à chaque itération et on obtient alors la méthode du gradient à pas variable. Parmi les nombreuses méthodes du gradient à pas variable, la plus populaire est la méthode du gradient à pas optimal.

*** Méthode du gradient à pas optimal**

La méthode du gradient à pas optimal est un algorithme de gradient dans lequel la taille de pas α_k est choisie pour atteindre le maximum de diminution de la fonction

objectif à chaque pas individuel. Plus précisément, un α_k est choisi pour minimiser

$f(x_k - \alpha \nabla f(x_k))$. En autre mots dit

$$\alpha_k = \arg\min_{\alpha \geq 0} f(x_k - \alpha \nabla f(x_k)) \tag{2.5}$$

Pour plus de détail voir [33,34].

2 Méthode de Newton

La méthode du gradient à pas optimal utilise uniquement les premières dérivées (gradients) pour sélectionner une direction de recherche appropriée. Cette stratégie n'est pas toujours la plus efficace. Si les dérivées supérieures sont utilisées, l'algorithme itératif résultant peut donner de meilleurs résultats que La méthode du gradient à pas optimal. La méthode de Newton (parfois appelée méthode de NewtonRaphson) utilise des première et seconde dérivées et donne même de meilleurs

résultats que La méthode du gradient à pas optimal si le point initial est proche du minimum. L'idée de cette méthode est la suivante. À partir d'un point de départ, nous construisons une approximation quadratique de la fonction objectif qui correspond aux valeurs de la première et la deuxième dérivées à ce point. Nous minimisons ensuite la fonction approximative (quadratique) au lieu de la fonction objectif d'origine. Nous utilisons l'optimum de la fonction approximative comme point de départ à l'étape suivante et répétons la procédure de manière itérative. Si la fonction objective est quadratique, alors l'approximation est exacte et la méthode donne le vrai optimum dans une étape. si, au contraire, la fonction objective n'est pas quadratique, alors l'approximation ne fournira qu'une estimation de la position du vrai optimum. Nous pouvons obtenir une approximation quadratique de la fonction objective deux fois continument différentiable $F : \mathbb{R}^n \to \mathbb{R}$

lisant le développement en série de Taylor de f autour du point actuel x_k, en négligeant les termes d'ordre plus grand ou égal 3. On obtient

$f(x) \backsimeq f(x_k) + (x - x_k)'\nabla f(x_k) + \frac{1}{2}(x - x_k)'F(x_k)(x - x_k) = q(x)$

$F(x_k)$ est la matrice hessienne.Soit x_{k+1} l'optimum de q , alors il vérifie $\nabla q(x_{k+1})$ =0 , d'où :

$\nabla q(x_{k+1}) = \nabla f(x_k) + F(x_k)(x - x_k) = 0$

Si la matrice$F(x_k)$ est définie positive, alors :

$$x_{k+1} = x_k - F(x_k)^{-1}\nabla f(x_k) \tag{2.6}$$

Cette formule récursive représente la méthode de Newton. Pour une étude approfondie de cette méthode et d'autres types de méthodes d'optimisation multidimensionnelle (méthode de quasi-Newton, méthode de gradient conjugué, l'algorithme DFP, l'algorithme BFGS,...) voir [33,34].

2.3 Méthodes méta-heuristiques

Une méta-heuristique est un algorithme d'optimisation visant à résoudre des

problèmes d'optimisation difficiles pour lesquels on ne connaît pas de méthode classique plus efficace. Les méta-heuristiques sont souvent inspirées par des systèmes naturels, qu'ils soient pris en physique (cas du recuit simulé [35]), en biologie de l'évolution (cas des algorithmes génétiques [36]) ou encore en éthologie (cas des algorithmes de colonies de fourmis ou de l'optimisation par essaims particulaires). Plusieurs ouvrages et articles ont été publiés sur ce sujet [37–40]. Dans la suite nous donnons quelques exemples de ce type des méthodes.

2.3.1 L'optimisation par essaims particulaires (OEP).

L'optimisation par essaims de particules (OEP) a été initialement introduite par Eberhart et Kennedy [41]. Le OEP est un algorithme de recherche basé sur la simulation du comportement social des oiseaux, des abeilles ou d'un bancs de poissons...etc [41]. Soit f : désignée la fonction objective à minimiser. L'essaim est défini comme un ensemble :$S = x_1, x_2, \ldots, x_N$, de N particules (solutions candidates). Chaque particule est caractérisée par :

1. $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})$: la position actuelle de la particule.

2. $v_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,n})$: Les particules sont supposées se déplacer dans l'espace de recherche A de manière itérative. Ceci est possible en ajustant les positions en utilisant ses vitesses v_i La vitesse est également adaptée de manière itérative pour rendre les particules capables de visiter potentiellement

toute région de A. La vélocité est réinitialisée en fonction des informations obtenues lors des étapes précédentes de l'algorithme. Ceci est implémenté en termes de mémoire, où chaque particule peut stocker la meilleure position qu'elle ait visitée au cours de sa recherche. Donc l'essaim S , conserve également un ensemble :$P = p_1, p_2, \ldots, p_N$

3. $p_i = (p_{i,1}, p_{i,2}, \ldots, p_{i,n})$: est la meilleure position visitée par la particule i. OEP est basé sur des modèles de simulation du comportement social ; ainsi, un mécanisme d'échange d'informations doit exister pour permettre aux particules de

communiquer mutuellement leur expérience. L'algorithme se rapproche d'optimum global avec la meilleure position visitée par toutes les particules. Par conséquent, il est judicieux de partager cette information importante.

4.$p_g(t)$: Soit g l'indice de la meilleure position dans P à une itération donnée t, c'est-à-dire

$$p_g(t) = \arg\min_i f(p_i(t)) \tag{2.7}$$

Alors, la première version de OEP est définie par les équations (1.8) et (1.9) [39] :

$$v_{i;j}(t+1) = v_{i;j}(t) + c_1 R_1(p_{i;j}(t) - x_{i;j}(t) + + c_2 R_2(p_{g;j}(t) - x_{i;j}(t)), \tag{2.8}$$

$$x_{i;j}(t+1) = x_{i;j}(t) + v_{i;j}(t+1) \tag{2.9}$$

où $i = 1; 2; ....; N$, $j = 1; 2; ....; n$ et t désigne l'itération en cours ; R1 et R2 sont des variables aléatoires uniformément réparties dans [0 ; 1] et c_1 ,c_2 sont des facteurs de pondération, également appelés paramètres cognitifs et sociaux, respectivement. À chaque itération, après la mise à jour et l'évaluation des particules, les meilleures positions (mémoire) sont également mises à jour. Ainsi, la nouvelle meilleure position de x_i

à l'itération t + 1 est définie comme dans l'équation (2.10) :

$$p_i(t+1) = \begin{cases} x_i(t+1) & si \quad f(x_i(t+1)) \leq f(p_i(t)), \\ p_i(t) & si \quad f(x_i(t+1)) > f(p_i(t)), \end{cases} \tag{2.10}$$

Le code de l'algorithme OEP est donné dans le tableau (2.1). Pour une étude approfondi sur cette algorithm voir [41].

Entrée :	nombre de particules N de l'essaim S ; les meilleures positions, P ..
Étape 1.	posé t = 0.
Étape 2.	Initialiser S et posez P = S .
Étape 3.	Évaluer S et P et définit l'indice g de la meilleure position.
Étape 4.	While (critère de résiliation non rempli).
Étape 5.	Mettre à jour S en utilisant les équations (2.8) et (2.9).
Étape 6.	Évaluer S .
Étape 7.	Mettez à jour P et redéfinit l'indice g.
Étape 8.	Poser t = t + 1.
Étape 9.	Fin While.
Étape 10.	Imprimer la meilleure position trouvée.

TABLE 2.1 – Le code de l'algorithme OEP.

2.3.2 Optimisation par Colonie d'abeilles

1 Les abeilles dans la nature

Comme les fourmis, les abeilles sont des insectes sociaux. Elles sont obligées de vivre en colonie très organisée, formée d'ouvrières, de faux-bourdon et d'une seule reine, et où chacune a un travail bien précis à faire. Les abeilles se nourrissent essentiellement de pollen et de miel. Elles vont butiner les fleurs pour prendre le nectar. Les abeilles adultes (âgées de 20 à 40 jours) deviennent habituellement des butineuses. Les abeilles butineuses jouent en général l'un des trois rôles suivants : butineuses actives, butineuses éclaireuses et butineuses inactives. Une colonie d'abeilles mellifères peut s'étendre sur de longues distances (plus de 10 km) et dans plusieurs directions simultanément pour exploiter un grand nombre de sources de nourriture. En principe, les abeilles visitent plus de parcelles de fleurs

contenant beaucoup de nectar ou de pollen qui peuvent être récoltés avec moins d'effort, alors que les parcelles de fleurs contenant moins de nectar ou de pollen doivent recevoir moins d'abeilles [42]. Dans le processus de recherche, l'emplacement de la source de nourriture représente la solution possible au problème, et la quantité du nectar et de pollen de cette source correspond à une valeur objective dite fitness.

Le processus de recherche de nourriture commence dans une colonie par l'envoi des butineuses éclaireuses aux différentes sources de nourriture. Les butineuses éclaireuses se déplacent au hasard d'un champ de fleurs à un autre.

Quand les butineuses éclaireuses (qui ont trouvé une source de nourriture et qui ont évalué (fitness) au-dessus d'un certain seuil de qualité (mesuré à la combinaison de certains constituants, tels que la teneur en sucre)) retournent dans la ruche, déposent leur nectar ou pollen et commencent à danser. Cette danse mystérieuse est essentielle à la communication entre les abeilles et contient trois informations concernant un source de nourriture : la direction dans laquelle il sera trouvé, sa distance par rapport à la ruche et son indice de qualité [42]. Ces informations aide la colonie à envoyer ses abeilles à cette source avec précision, sans utiliser de guides. Après avoir dansé, les danseuses (c'est-à-dire les butineuses éclaireuses) retournent au source de nourriture avec des abeilles qui attendaient à l'intérieur de la ruche. Plus d'abeilles sont envoyées dans des sources plus prometteuses. Cela permet à la colonie de rassembler les aliments rapidement et efficacement. Lors de la récolte dans une source, les abeilles surveillent le niveau de nourriture. Cela est nécessaire pour choisir le type de la prochaine danse à leur retour dans la ruche.

2 Algorithme d'abeilles

L'algorithme des colonies d'abeilles [43] est une méthode basée sur la population pour trouver une solution optimale aux problèmes de recherche. Elle est inspirée du comportement des abeilles dans la nature [43,44]. L'algorithme néces-

site la définition d'un certain nombre de paramètres :

1. n : nombre de butineuses éclaireuses,
2. m :nombre de sources sélectionnées sur n sources visitées,
3. e : nombre de meilleurs sources sur m champs sélectionnés,
4. nep : nombre d'abeilles recrutés pour les e meilleurs sources,
5. nsp : nombre d'abeilles recrutées pour les autres sources sélectionnées,
6. ngh : taille initiale des sources,
7. critère d'arrêt.

Le tableau 3.2 montre le code de l'algorithme dans la plus simple forme. À l'étape 4, les abeilles qui ont les plus grand fitness sont choisies comme « abeilles sélectionnées » et les sources visités par celles-ci sont choisies pour une recherche au voisinage. Ensuite, aux étapes 5 et 6, l'algorithme effectue des recherches au voisinage des sources sélectionnées, affecter plus d'abeilles pour rechercher près de e meilleurs sources. À l'étape 7, les abeilles restantes de la population sont affectées

Étape 1.	Initialiser la population avec des solutions aléatoires.
Étape 2.	Évaluer la fitness de la population .
Étape 3.	While (critère d'arrêt non rempli) Formation d'une nouvelle population.
Étape 4.	Sélectionnez les sources pour une recherche au voisinage.
Étape 5.	Recrutez des abeilles pour les sources sélectionnés (plus d'abeilles pour les meilleurs sources) et évaluer les fitness.
Étape 6.	Sélectionnez l'abeille correspond à max(fitness) de chaque source.
Étape 7.	Affecter aux abeilles restantes une recherche aléatoire et évaluer leurs fitness.
Étape 8.	Fin While.

TABLE 2.2 – Le code de l'algorithme d'abeilles.

de manière aléatoire autour de l'espace de recherche à la recherche de nouvelles solutions potentielles. Ces étapes sont répétées jusqu'à ce qu'un critère d'arrêt soit rempli.

Chapitre 3

Applications à l'optimisation

3.1 Introduction à l'algorithme d'optimisation de la mouche des fruits

3.1.1 Calculs évolutifs et intelligence artificielle

Le calcul évolutif est un nom commun qui fait référence à la " survie du plus apte et à l'élimination de l'inapte " de la théorie darwinienne. des plus aptes et à l'élimination des inaptes" de la théorie darwinienne. concept, le processus évolutif de la nature est pratiquement simulé pour établir des établir des modes de calcul, tels que les algorithmes génétiques du Prof. Holland à ses débuts [46]. Cependant, plus récemment, le cœur de l'évolution de l'évolution a commencé à être détourné vers le comportement de recherche de nourriture des animaux et le comportement de groupe, comme les algorithmes de particules. comportement de groupe, comme l'optimisation par essaims de particules (OEP) du Prof. Eberhart et l'Artificial Fish Swarm Algorithm (AFSA) proposé par le Prof. Li de la Chine continentale

[47].

Ces deux algorithmes sont développés à partir des comportements de recherche de nourriture des populations animales. Cet auteur (Wen-Tsao Pan, 2011) cet auteur s'est inspiré du comportement de butinage de la drosophile et a drosophile et a proposé l'algorithme d'optimisation de la mouche à fruits. communauté académique, afin que nous puissions réaliser ce nouvel algorithme pratique recherche, augmenter la chance que nous soumettons à la SCI internationale, SSCI revues acceptées.

3.1.2 Le concept de base de l'algorithme d'optimisation de la mouche des fruits

L'algorithme d'optimisation de la mouche des fruits a été inventé par le professeur Pan, un universitaire de Taiwan . Pan, un universitaire de Taiwan . Il s'agit d'une nouvelle méthode de déduction de l'optimisation globale basée sur le comportement de la mouche à fruits. sur le comportement de recherche de nourriture de la mouche à fruits. La perception sensorielle de la mouche à fruits perception sensorielle de la mouche des fruits est meilleure que celle des autres espèces, en particulier l'odorat et la vision. la vision. L'organe olfactif de la drosophile peut capter diverses odeurs de l'air, et même une source de nourriture. l'air, et même une source de nourriture à 40 km de distance. Ensuite, la mouche à fruits vole jusqu'à la source de nourriture. nourriture, utilise sa vision aiguë pour trouver la nourriture et où ses congénères se rassemblent, et puis elle vole dans cette direction, comme le montrent les figures 3.1 et 3.2.

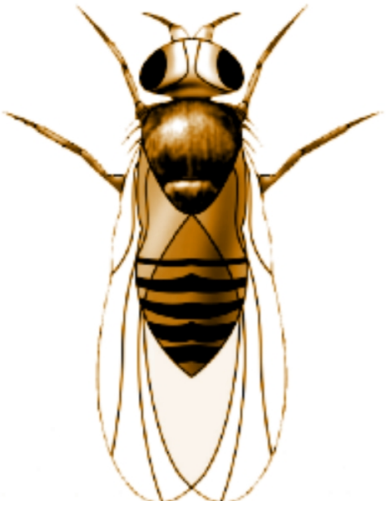

FIGURE 3.1 – *Schéma de la structure du corps de la drosophile.*

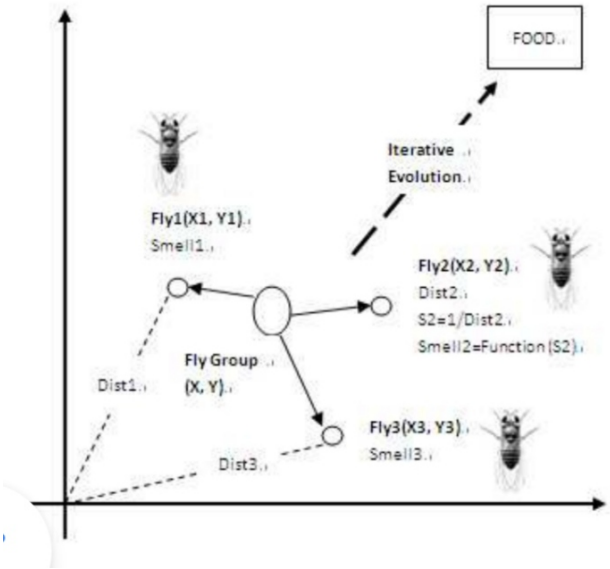

FIGURE 3.2 – *Schéma de la recherche itérative de nourriture de la mouche à fruits essaim.*

Les caractéristiques de la mouche à fruits en matière de recherche de nourriture

sont réduites à quelques étapes nécessaires et des exemples de procédures, à titre de référence pour les lecteurs. Les étapes sont décrites comme suit : La position initiale aléatoire d'un essaim de mouches à fruits est illustrée à droite dans la de la figure 1.

Init $X-axis$; $InitY-axis$,

- Direction et distance aléatoires de la recherche de nourriture en utilisant le sens de l'odorat d'un individu mouche à fruits. l'odorat d'un individu mouche à fruits.

$X_i = X-axis + RandomValue$,

$Y_i = Y-axis + RandomValue$,

- Comme l'emplacement de l'aliment ne peut être connu, la distance (Dist) à l'origine est estimée avant la valeur de décision de la concentration olfactive (S). l'origine est estimée avant la valeur de décision de la concentration de l'odeur (S). est calculée ; cette valeur est l'inverse de la distance.

$Dist_i = \sqrt{X_i^2 + Y_i^2}$;$S_i = \dfrac{1}{Dist_i}$,

- La valeur de décision de concentration d'odeur (S) est substituée dans la fonction de décision de fonction de décision de concentration d'odeur (également connue sous le nom de fonction Fitness) pour pour calculer la concentration d'odeur (Smelli) dans la position de l'individu de la mouche des fruits. individu.

$Smell_i = FOnction(S_i)$,

- Déterminer la mouche à fruits ayant la concentration maximale d'odeur parmi l'essaim de mouches à fruits (recherche de la valeur maximale)

$[bestSmellbestIndex] = max(Smell)$,

- Retenir la meilleure valeur de concentration d'odeur et les coordonnées x, y, ici la essaim de mouches à fruits vole vers la position par vision.

$Smellbest = bestSmell$,

$X-axis = X(bestIndex)$,

$Y-axis = Y(bestIndex)$,

- Entrer dans l'optimisation itérative, répéter les étapes d'exécution 2-5, et juger si la concentration d'odeur est meilleure que la concentration itérative précédente.

concentration d'odeur, si oui, exécutez l'étape 6.

Algorithm 1. Chaotic FOA pseudocode

//Algorithm initialization Set the population size PS and maximum number of iterations I_{max} //Initialize fruit fly swarm location in the search space n For $i = 1, .., PS.\ x_{i,j} = lower_boun + (upper_bound - lower_bou) \times rand()\ ; j = 1, ..., n.$

End For

$\triangle \leftarrow arg(\min f(X_i))\ ; i = 1, 2, ..., PS.$

Set swarm location

// Set optimal solution and iteration counter :

$X^* = \triangle,$

$Iter = 0$ Repeat //Smell-based (osphresis) foraging phase For $j = 1, \ldots, PS.$ // Generate food source $X_i = (x_{i,1}, x_{i,2}, ...x_{i,n},),$

$alpha = chaos()$ //Determine chaotic parameter

$x_{i,j} = x_{i,j} + alpha(x_{i,j} - x_j^*)\ ; i = 1, 2, ..., PS\ ; j = 1, 2, ..., n.$

// Limit the result

If $x_{i,j} > upper_boundthen,$

$x_{i,j} = upper_bound,$

End If

End For

//Vision-based foraging phase $X_{best} = arg(\min f(X_i)),\ i = 1, 2, ..., PS$

//Find global best solution

If $f(X_{best}) < f(\triangle)$then

$\triangle = X_{best},$

End If

If $f(\triangle) < f(X^*)$then

$X^* = \triangle$,

End If

Until the maximum number of iteration is reached :

$Iter = Iter_{max}$,

Tableau 1

Cartes chaotiques utilisées dans cette étude.

N	Nom de la carte	Équation
1	Chebyshev	$X_{i+1} = \cos(i \cos^{-1}(x_i))$
2	Iterative	$X_{i+1} = \sin(\frac{a\pi}{x_i}), a = 0,7$
3	Logistique	$X_{i+1} = ax_i(1 - X(i)), a = 4$

Les performances de l'algorithme sont testées sur troi fonctions non linéaires différentes.

3.2 Algorithme d'optimisation chaotique de la mouche du fruit

Cette section présente un nouvel algorithme d'optimisation de la mouche à fruits en introduisant un nouveau paramètre amélioré par le chaos. Nous commençons l'explication de l'algorithme d'optimisation chaotique de la mouche du fruit chaotique (CFOA) comme suit.

3.2.1 Initialisation de l'algorithme

L'emplacement initial de l'essaim peut avoir une influence majeure sur la vitesse de convergence et le résultat final. Comme première amélioration, CFOA détermine l'emplacement initial de l'essaim de mouches à fruits en choisissant la meilleure solution parmi les solutions générées aléatoirement par PS. meilleure parmi les solutions générées aléatoirement par PS. De manière similaire à ce qui est ce qui a été trouvé dans [46], ce calcul de la position initiale de l'essaim entraîne une une convergence plus rapide et une meilleure solution de l'algorithme à la fin de l'expérience. l'algorithme à la fin du cycle expérimental.

3.2.2 FOA du chaos

Dans le FOA, l'influence cruciale sur les performances de l'algorithme concerne le calcul des sources de nourriture. L'implémentation de base de cette technique métaheuristique suppose la randomisation des variables $X_{i;j}$ en utilisant une distribution uniforme. Ce n'est souvent pas un bon choix, surtout lorsqu'il s'agit de problèmes complexes, non linéaires et multimodaux. Afin d'améliorer la convergence et la vitesse globale de FOA, nous introduisons un nouveau paramètre, alpha. qui est utilisé pour la génération de sources d'alimentation. En particulier, nous avons modifié l'équation $X_{i;j} = \delta \pm rand()$,$j = 1, ..., n$ de sorte qu'il implique une variable chaotique comme suit : $x_{i,j} = x_{i,j} + alpha(x_{i,j} - x_j^*)$,$i = 1, ..., PS.$,$j = 1, ..., n.$ (en utilisant MATLAB).

où x_j^* est la meilleure solution actuelle. De cette manière, nous forçons les individus à se diriger vers la meilleure solution optimale jusqu'à présent de manière chaotique. Cela s'est avéré être un énorme avantage en comparaison avec la FOA de base et la FOA avec distribution de Levy. La procédure complète d'un nouveau CFOA est présentée à la Algorithm 1(choatique FOA pseudocode) , nous étudions l'influence de trois cartes chaotiques différentes, unidimensionnelles et non-inversibles, de la même manière que d'autres études récentes [47]. . Une description mathématique et une présentation graphique de ces cartes pour 300 itérations sont données dans le Tableau 1 , respectivement. Il est important de noter que le comportement chaotique Chaque carte chaotique présentée dans la Fig. 3.3 a un point de départ de 0,7. point de départ de 0,7. Les cartes qui ne produisent pas de valeurs dans la plage de [0,1] sont normalisées pour correspondre à cette échelle.

3.3 Détails de la mise en œuvre

3.3.1 Fonctions de test

La FOA chaotique présentée dans cette étude est évaluée à l'aide de 4 différentes fonctions bien connues [48]. Le site description mathématique et la présentation graphique de ces fonctions sont données dans leTableau 2 . Pour tous les problèmes de test, l'optimum global est égal à $f(x_j^*) = 0$

Les limites des fonctions sont égales à leurs plages initiales connues. Après chaque itération de l'algorithme, la contrainte de frontière pour chaque $x_{i,j}$ est appliquée.

3.3.2 Critère de réussite

En plus des mesures habituelles pour l'évaluation des algorithmes, telles que le meilleur, la moyenne et la médiane, nous appliquons également dans ce document le critère du taux de réussite. Le paramètre de taux de réussite S_r est défini comme [48] :
$S_r = \frac{N_{success}}{N_{all}} \times 100.$
où $N_{success}$ est le nombre d'essais réussis, etNallest le nombre d'essais. Comme dans d'autres études , un essai expérimental est considéré comme est considérée comme réussie si la solution finale de l'algorithme est proche de l'optimum recherché. Le critère de proximité dépend de l'espace de recherche espace de recherche d'une fonction particulière, et est défini comme suit [48] :

$$\mid X^{gbest} - X^* \mid \leq (upper_bound - lower_bound) \times 10^{-4},$$

est le meilleur résultat global obtenu par l'algorithme développé l'algorithme développé.

3.3.3 Études de test et initialisation

Dans ce document, nous avons testé chaque fonction avec 50 exécutions indépendantes de l'algorithme. exécutions indépendantes de l'algorithme. Les conditions initiales de chaque test sont complètement différentes, de sorte que le résultat de l'algorithme est pratiquement indépendant de la la position de départ de l'essaim de mouches à fruits. Pour évaluer complètement la performance du CFOA, nous avons utilisé des mesures statistiques telles que les valeurs objectives médianes et moyennes, ainsi que leurs écarts types. Ces informations sont fournies pour chaque carte chaotique et chaque fonction testée.

De plus, des études approfondies concernant les réglages des paramètres sont réalisées. À partir des expériences menées, nous avons conclu qu'une population de 50 individus et 700 itérations par cycle expérimental est suffisante pour tous les cas de test. De même, dans toutes les expériences, la valeur initiale de 0,95 pour le paramètre "ph" s'est avérée être un bon choix.

3.4 Études expérimentales

3.4.1 Résultats de calcul du CFOA sur des problèmes de référence

Les résultats de calcul pour toutes les fonctions utilisant toutes les cartes sont donnés dans les Tableaux 3-6 (le meilleur résultat est indiqué en caractères gras). La FOA chaotique est testé sur 4 problèmes de référence en utilisant les fonctions Chebyshev, itérative et logistique . En plus des mesures statistiques mentionnées, nous avons ajouté les meilleurs et les pires résultats obtenus sur 50 exécutions indépendantes, ainsi que le temps moyen pour un tel test.

Globalement, les résultats expérimentaux prouvent l'utilité de la implémentation du chaos dans la FOA. Dans chaque cas testé, la sortie finale du CFOA était très proche de la fonction optimale souhaitée. Pour trois sur les quatre fonctions, la meilleure valeur minimale est obtenue avant que le nombre maximum d'itérations ne soit atteint. Nous fournissons également des informations supplémentaires sur l'itération exacte dans laquelle la fonction optimale est atteinte

Les expériences montrent que les cartes de Chebyshev, Iterative, Logistique, donnent les meilleurs résultats dans tous les cas (tableaux 3-6). En particulier, ces cartes ont atteint la performance maximale de l'algorithme avant la fin d'un cycle expérimental, la carte de Tchebychev donne les meilleurs résultats en termes de convergence la plus rapide de l'algorithme.

Tableau 2

ID F	Function	name Equation	Upper and lower bound	Dimension n	Type
F1	Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	Ub=100 Ul=-100	30	Unimodal
F2	Sum squares	$f(x) = \sum_{i=1}^{n} ix_i^2$	Ub=10 Ul=-10	30	Unimodal
F3	Rosenbrock	$f(x) = \sum_{i=1}^{n-1}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	Ub=30 Ul=-30	30	Unimodal

Tableau 3

Résultats du CFOA pour la fonction Sphere (F1) après 50 exécutions indépendantes.

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time (s)
1	Chebyshev map	$< 1E-309(Iter.540)^a$	<1E-309	<1E-309	<1E-309	N/A	0.5244
2	Iterative map	$< 1E-309(Iter.659)^a$	<1E-309	<1E-309	<1E-309	N/A	0.4323
3	Logistic map	$< 1E-309(Iter.630)^a$	<1E-309	<1E-309	<1E-309	N/A	0.4478

(a) Itération au cours de laquelle le meilleur résultat est obtenu.

Tableau 4

Résultats du CFOA pour la fonction Sum squares(F2) après 50 exécutions indépendantes.

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time (s)
1	Chebyshev map	$< 1E-308(Iter.539)^a$	<1E-308	<1E-308	<1E-308	N/A	0.5009
2	Iterative map	$< 1E-308(Iter.655)^a$	<1E-308	<1E-308	<1E-308	N/A	0.4124
3	Logistic map	$< 1E-308(Iter.628)^a$	<1E-308	<1E-308	<1E-308	N/A	0.4255

(a) Itération au cours de laquelle le meilleur résultat est obtenu.

Tableau 5

Résultats du CFOA pour la fonction Quartic (F3) après 50 exécutions indépendantes.

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time (s)
1	Chebyshev map	8.7628 E-006	2.2289 E-004	2.4749 E-003	3.8616 E-002	6.1081E-003	0.6125
2	Iterative map	1.8656 E-007	1.9228E-005	2.4341E-005	1.114E-004	2.2242E-005	0.64
3	Logistic map	4.6066E-007	1.9228E-005	2.4341E-005	1.114E-004	2.3347E-005	0.6301

(a) Itération au cours de laquelle le meilleur résultat est obtenu.

Tableau 6

Résultats du CFOA pour la fonction Rastrigin (F4) après 50 exécutions indépendantes.

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time (s)
1	Chebyshev map	$< 1E-14(Iter.28)^a$	<1E-14	<1E-14	<1E-14	N/A	0.4356
2	Iterative map	$< 1E-308(Iter.34)^a$	<1E-14	<1E-14	<1E-14	N/A	0.4264
3	Logistic map	$< 1E-308(Iter.32)^a$	<1E-14	<1E-14	<1E-14	N/A	0.4187

(a) Itération au cours de laquelle le meilleur résultat est obtenu.

que c'est le meilleur choix dans tous les cas de test. De manière similaire à l'expérience présentée ci-dessus, nous avons testé ces algorithmes sur trois problèmes de référence sélectionnés avec 50 exécutions indépendantes.

CONCLUSION GÉNÉRALE

L'objectif de ce travail était d'étudier l'Algorithme d'optimisation chaotique de la mouche des fruits .

Nous avons divisé notre travaille en trois chapitres.
Le premier chapitre est consacré a la présentation de notions générales sur les systèmes dynamiques comme les notions de base :la définition de système dynamique discret ,points fixes , la théorie de bifurcation et chaos en outre nous avons présenté des exemples .

Nous avons abordé dans le deuxième chapitre une introduction à l'optimisation :Types d'optima, Méthodes d'optimisation classiques (Optimisation unidimensionnelle,Optimisation multidimensionnelle),Méthodes méta-heuristiques (L'optimisation par essaims particulaires (OEP),Optimisation par Colonie d'abeilles)

Dans Le troisième chapitre on a donné une étudie de l'Algorithme d'optimisation chaotique de la mouche des fruits .

Bibliographie

[1] A. Boukabou, *Méthode de contrôle des systèmes chaotique d'ordre élevé et leur application pour la synchronisation : Conribution à l'élaboration de nouvelles approches, Thèse de Doctorat Univ. Mentouri*,Constantine 1, 2006.

[2] *E. N. Lorenz, Deterministic non-periodic flow. Journal of the Atmospheric Sciences, 20 :130–141, 1963.*

[3] *T. Y. Li et J. A. Yorke, Period therre implies chaos. American Mathematical Monthly, 82 :985-992, 1975.*

[4] *J. Liu, S. J. Li, New hybrid conjugate gradient method for unconstrained optimization, Appl. Math. Comput. 245 (2014) 36-43.*

[5] *TW.C. Chen, V. S. Vassiliadis, Solution of general nonlinear optimization problems using the penalty/ modified barrier method with the use of exact hessians, Comput. Chem. Eng. 27(4) (2003) 501-525.*

[6] *J. A.T. Machado, Optimal tuning of fractional controllers using genetic algorithms, Nonlinear Dyn. 62(12)(2010) 447-452.*

[7] *D.Bunnag , M. Sun, Genetic algorithm for constrained global optimization in continuous variables, Appl. Math. Comput.171(1)(2005) 604-636.*

[8] *R. Bououden, M-S. Abdelouahab, On Efficient Chaotic Optimization Algorithm Based on Partition of Data Set in Global Research Step . Nonlinear Dynamics and Systems Theory. 18 (1) (2018) 42-52.*

[9] J. Holmes, P. Nonlinear Osciators, *Dynamical Systèmes, and Bifurcations of Vector Fields*, Guckenheimer, Mathematical Sciences. Springer verlag édition, 1983.

[10] J.K. Hale, H. Kocak, *Dynamics and Bifurcations.* Applied Mathematics, Publisher SpringerVerlag New York, 1991.

[11] H. Dang-Vu,C. Delcarte, *Bifurcations et Chaos.* Paris, Ellipses, 2000.

[12] Y. Kuznetsov, *Elements of Applied Bifurcation Theory.* Springer-Verlag New York, 2004.

[13] T. Yoshizawa, *Theory and the Existence of Periodic Solutions and Almost Periodic Solutions. Applied Mathematical Sciences Series,*Publisher Springer-Verlag New York, 1975.

[14] M. Lakshmanan, S. Rajaseekar, *Nonlinear Dynamics Integrability, Chaos and Patterns. Advanced Texts in Physics,*Publisher Springer-Verlag Berlin Heidelberg, 2003

[15] S. Wiggins, *WIntroduction to Applied Nonlinear Dynamical Systems and Chaos, Texts in Applied Mathematics,*Springer-Verlag New York, 2003.

[16] R.L. Devaney, *An introduction to chaotic dynamical systems.In Adissionwisley,*Redwood City, CA 37.

[17] *C. Mira, L. Gardini,A. Bugola,J-C. Cathala, Chaotic dynamics in two-dimensional noninvertible maps. World Scieniific, 1996.*

[18] *K.T. Alligood, T.D. Sauer, et J.A. Yorke, Chaos : an Introduction to Dynamical Systems, Springer-Verlag édition, 1996.*

[19] *M. Rosenstein, J. Collins,C. Deluca, (1993), "A practical method for calculating largest Lyapunov exponents for small data sets," Physica, Vol. 65, pp. 117-134.*

[20] *A.Wolf, J. Swift,H. Swinney, J. Vastano, (1985), "Determining Lyapunov exponents from a time series," Physica, Vol. 16, pp. 285-317.*

[21] *T.S. Parker, L.O. Chua,Pratical Numerical Algorithms For Chaotic Systems. Edition Springer-Verlay, 1989.*

[22] *A.J. Chorin, (1981), "Estimates of intermittency, Spectra and blow-up in devloped turbulence," Comm. Pure Appl. Math., Vol. 4, pp. 853.*

[23] *T-Y. Li, J.A. York, (1975), "Period three implies chaos," Amer. Math. Mon., Vol. 82, pp. 985-992.*

[24] *J. Guckenheimer, P. Holmes, Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields, Springer-Verlag, New York, 1986.*

[25] *C. Sparow, (1981), "Chaos in three dimensional single loop feedback systems with a piecewise linear feedback function," J. Math. Anal. Appl., Vol. 83.*

[26] *C.Grebogi, E. OTT, S. Pelikan, J.A. Yorke, (1984), "Strange attractors that are not chaotic," Physica D, Vol. 13, pp. 261-268.*

[27] *G.B. Mindlin, X-J. Hou, H.G. Solari, R. Gilmore,N.B. Tufilaro, (1990), "Classification of strange attractors by integers", Phys. Rev. Lett., Vol. 64, pp. 2350.*

[28] *M. Hénon, (1976), "A Two Dimentionnal Mapping with a Strange Attractor," Comm. Math. Phys. Vol. 50, pp.69-76.*

[29] *M. Itoh,T. Yang,L.O. Chua, (2001), "Conditions for impulsive synchronization of chaotic and hyperchaotic systems," Int. J. Bifurcation Chaos Appl. Sci. Eng., Vol. 11, pp. 551-8.*

[30] *D.L. Hitzl,F. Zele, (1985), "An exploration of the Hénon quadratic map," Physica D, Vol.14(3), pp. 305–326.*

[31] *Z.Y,Yan (2006), "Q-S (Complete or Anticipated) Synchronization Backstepping Scheme in a Class of Discrete-Time Chaotic (Hyperchaotic) Sys-*

tems : A Symbolic-Numeric Computation Approach," Chaos, Vol. 16, pp. 013119-11.

[32] *X.Y.Wang, Chaos in Complex Nonlinear Systems, Publishing House of Electronics Industry, Beijing, 2003.*

[33] *E.K.P.Chong, S.H. Zak, An Introduction to Optimization. Wiley, 2001.*

[34] *S. Rao, Engineering Optimization Theory and Practice. Wiley, 2009.*

[35] *R. Chibante, Simulated Annealing, Theory with Applications. Springer Netherlands, 1987.*

[36] *D.E. Goldber, Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional, 1989.*

[37] *T. El-Ghazali, Metaheuristics : from design to implementation. Wiley, 2009.*

[38] *K.F, Doerner, M. Gendreau, P. Greistorfer, W. Gutjahr, R.F. Hartl, M. Reimann, Metaheuristics : Progress in Complex Systems Optimization. Springer, 2007.*

[39] *T. Ibaraki, K. Nonobe, M. Yagiura, Metaheuristics : Progress as Real Problem Solvers. Springer, 2005.*

[40] *J. Dréo, A. Pétrowski, P. Siarry, E. Taillard, Metaheuristics for Hard Optimization : Methods and Case Studies. Springer, 2005.*

[41] *J. Kennedy and R. Eberhart, Particle Swarm Optimization, Proceedings of IEEE International Conference on Neural Networks. IV(1995) 1942–1948.*

[42] *K.E. Parsopoulos, M.N. Vrahatis, Particle Swarm Optimization and Intelligence : Advances and Applications. Information Science Publishing, 2010.*

[43] *E, Bonabeau, M, Dorigo and G, Theraulaz, Intelligence : from Natural to Artificial Systems. Oxford University Press, New York, 1999.*

[44] *D.T. Pham, E. Koc, A. Ghanbarzadeh, S. Otri, S. Rahim, M. Zaidi, The Bees Algorithm–a novel tool for complex optimisation problems, Proceedings of the Second International Virtual Conference on Intelligent Production Machine and Systems. (2006) 454–461.*

[45] *D.T. Pham, M. Castellani, The Bees Algorithm : modelling foraging behaviour to solve continuous optimization problems, Proceeding of Institute Mechanical Engineering, C : Journal of Mechanical Engineering and Science, 223(2009) 2919–2938.*

[46] *E. I.Altman, 1968, Financial Ratios, DiscriminantAnalysis.*

[47] *S. Saremi, S. Mirjalili, A. Lewis, Biogeography-based optimisation with chaos, Neural Comput. Appl. 2014,http ://dx.doi.org/10.1007/s00521-014-1597-x .*

[48] *A.H, Gandomi, X.S, Yang, S. Talatahari, A.H. Alavi, Firefly algorithm with chaos, Commun. Nonlinear Sci. Numer. Simul. 18 (1) (2013) 89–98.*